

# A Users Guide for the NASA ANOPP Propeller Analysis System

---

*L. Cathy Nguyen and Jeffrey J. Kelly*  
*Lockheed Engineering & Sciences • Hampton, Virginia*

Printed copies available from the following:

NASA Center for AeroSpace Information  
800 Elkridge Landing Road  
Linthicum Heights, MD 21090-2934  
(301) 621-0390

National Technical Information Service (NTIS)  
5285 Port Royal Road  
Springfield, VA 22161-2171  
(703) 487-4650

# INTRODUCTION TO ANOPP-PAS

## Summary

The purpose of this report is to document improvements to the Propeller Analysis System of the Aircraft Noise Prediction Program (PAS-ANOPP) and to serve as a users guide. An overview of the functional modules and modifications made to the Propeller ANOPP system are described. Propeller noise predictions are made by executing a sequence of functional modules through the use of ANOPP control statements. The most commonly used ANOPP control statements are discussed with detailed examples demonstrating the use of each control statement. Originally, the Propeller Analysis System included the angle-of-attack only in the performance module. Recently, modifications have been made to also include angle-of-attack in the noise prediction module. A brief description of PAS prediction capabilities is presented which illustrate the input requirements necessary to run the code by way of ten templates. The purpose of the templates are to provide PAS users with complete examples which can be modified to serve their particular purposes. The examples include the use of different approximations in the computation of the noise and the effects of synchrophasing. Since modifications have been made to the original PAS-ANOPP, comparisons of the modified ANOPP and wind tunnel data are also included. Two appendices are attached at the end of this report which provide useful reference material. One appendix summarizes the PAS functional modules while the second provides a detailed discussion of the TABLE control statement.

# TABLE OF CONTENTS

	Page
1. Introduction .....	1
2. Information Resources .....	3
3. Module Documentation .....	4
4. Control Statements .....	7
4.1 Single Directive Control Statements	
4.1.1 ANOPP .....	8
4.1.2 STARTCS .....	8
4.1.3 LOAD .....	8
4.1.4 UNLOAD .....	8
4.1.5 PARAM .....	8
4.1.6 EVALUATE .....	9
4.1.7 EXECUTE .....	10
4.1.8 ENDCS .....	10
4.2 Multiple Directive Control Statements	
4.2.1 UPDATE .....	11
4.2.2 TABLE .....	13
5. Example Programs	
5.1 Generate Module Documentation .....	16
5.2 Generate Atmospheric Data .....	16
5.3 Geometry Module Demonstration .....	17
6. Module Update	
6.1 Propeller Performance Module Modification .....	19
6.2 Subsonic Propeller Noise Module Modification .....	20
7. Description of PAS Predictions .....	26
8. PAS Program Templates	
8.1 Blade Geometry	
8.1.1 Improved version of PAS .....	28
8.1.2 Old version of PAS .....	32
8.2 Prediction of Performance and Loads	
8.2.1 Execute the Performance Module without Iteration .....	38
8.2.2 Execute the Performance Module with Iteration .....	39
8.2.3 Use PAS Loads for Input .....	42
8.2.4 Experimental Loads for Input .....	42

8.3	Near-Field Noise Prediction .....	45
8.4	Noise Bubble for Far-Field Noise Prediction .....	47
8.5	Flyover Noise Prediction	
8.5.1	One Propeller .....	49
8.5.2	Tilt Rotor .....	51
9.	PAS Prediction and Measured Data	
9.1	Results of PAS Studies	
9.1.1	Four Methods from SPN .....	58
9.1.2	Synchrophasing Using PAS .....	58
9.2	Comparison with Measured Data .....	60
10.	References .....	62
Appendix A	- Functional Modules .....	63
Appendix B	- TABLE Control Statement Discussion .....	64



## 1. Introduction

This document serves as an introduction and guide to the Aircraft Noise Prediction Program executive system and the Propeller Analysis System (PAS). Elements of this report such as the executive overview and module documentation are analogous to reference 1. This report is written for the user who is interested in making propeller noise predictions on work stations or on main frame computers in batch mode.

It is beneficial for users to understand some of the ANOPP program concepts to be discussed later in the manual. The ANOPP System is divided into two parts, the Executive System and the Functional Module Library. A hierarchical representation of ANOPP components is shown in figure 1. The Executive System controls execution of ANOPP and consists of several managing routines and a group of general utilities. The purpose of each major element in the Executive System is listed below

- The Executive Manager controls execution of ANOPP controls statements.
- The Data Base Manager controls activities of data tables and data members.
- The Dynamic Storage Manager allows core sharing and dynamic dimensioning of variable arrays.
- The General Utilities provide access to interpolation routines and other general functions.

More information concerning the Executive System can be found in reference 2. The Functional Module Library contains all the subprograms which perform noise prediction functions.

A flow chart of the ANOPP-PAS system is shown in figure 2. The theory for PAS is documented in reference 3. To make a propeller noise prediction using the ANOPP-PAS system, several function modules must be executed in a defined sequence. The procedure begins by choosing between the original and the improved PAS modules. Originally, PAS consisted of the Rotating Blade Shape module (RBS), the Blade Section Aerodynamics module (RBA), and the Blade Section Boundary-Layer module (BLM). New modules were created to ease inputting the blade geometry and to provide additional compressibility correction options. In the improved version of PAS, the first letter of each module was changed to I such as IBS, IBA, IBL. It is suggested that the improved modules of PAS be used. For an explanation of the improved and modified PAS see reference 4.

The next step in the procedure is to determine the propeller performance. The Propeller Performance Module (PRP) may be executed several times until the pitch has converged. If the pitch does not converge, the ANOPP run will be stopped. Otherwise the next step is to compute the propeller loads using the Propeller Loading Module (PLD). The

last step is to calculate the propeller noise using the three noise prediction modules which are the Subsonic Propeller Noise (SPN), the Transonic Propeller Noise (TPN), and the Propeller Trailing Edge Noise (PTE) Modules.

PAS allows predictions to be made in several reference frames. For wind tunnel noise predictions, modules one to six are executed. For flyover noise predictions, modules one to fourteen are executed. The Atmospheric Module (ATM) and the Atmospheric Absorption Module (ABS) build the atmospheric table. The flight path is defined by the Steady Flyover Module (SFO). The Geometry Module (GEO) computes the range and directivity angles from observer to the noise source. The Tone Propagation module (PRT) propagates the tone noise from the tone noise modules SPN and TPN and the Broadband Propagation module (PRO) propagates the broadband noise from the PTE module. The Noise Level Module (LEV) sums the noise, computes overall sound pressure level (OASPL), A-weighted sound pressure level, D-weighted sound pressure level, perceived noise level (PNL), and tone-corrected perceived noise level (PNLT). Effective Noise Module (EFF) computes effective perceived noise level (EPNL) and sound exposure level (SEL). A summary of the ANOPP PAS functional modules can be found in Appendix A.

Section 2 contains information resources that can be of aid to users. Module documentation with examples containing informative comments are the subject of Section 3. Section 4 describes the eleven most often used ANOPP control statements which will enable the user to set up and execute any ANOPP module. Three examples are provided in Section 5 which show how to set up a PAS prediction. Section 6 provides a summary of the improved and updated PAS (third version) which incorporates angle-of-attack in the noise prediction. A brief description of the capabilities and options of PAS is presented in Section 7. Section 8 contains ten templates to assist users in building a blade geometry table, an aerodynamics table such as lift and drag, and to compute the performance and the loads. Templates for the wind tunnel noise prediction and for the flyover noise prediction are included. Several templates are provided as examples to help users build a job for particular purposes. Results of the studies using PAS and the comparison of the measured data with PAS predictions are shown in Section 9.

The appendices provide supplemental information which will be useful as reference material. Appendix A is a summary of the functional modules provided in tabular format. Included in this table is the full title for each module, the associated ANOPP abbreviation, and a brief description of the function of that module. Appendix B includes a more in-depth discussion of the TABLE control statement.



## 2. Information Resources

Five manuals are available for users to obtain more information about PAS. The first document is the Aircraft Noise Prediction Program User's Manual (ref. 5) which contains a detailed explanation of the ANOPP executive system. The second document is the Aircraft Noise Prediction Program Theoretical Manual, Part 1 which contains the propagation and atmospheric absorption models (ref. 6). The third document is Part 3 of the Aircraft Noise Prediction Program Theoretical Manual which contains the propeller analysis theories, reference 3. The fourth document is the NASA Aircraft Noise Prediction Program Improved Propeller Analysis System (ref. 4). This manual describes the modifications and improvements that were made to the propeller analysis system. For the user who is interested in making propeller noise predictions without angle-of-attack on an IBM-PC, the Aircraft Noise Prediction Program Propeller Analysis System IBM-PC Version User's Manual (ref. 7) is available.

### 3. Module Documentation

User documentation is maintained as a preface to the FORTRAN source code. This is done to ensure that the correct documentation is available for each version of the program in existence. This documentation is maintained on line and is accessible to the user.

Figure 3 shows the format for the documentation of each module. The most important descriptors to the user are the INPUT, OUTPUT, and DATA BASE STRUCTURE. Under INPUT and OUTPUT, there are user parameters and unit members. A user parameter retains its value for each execution of a module. A unit member is closely related to a file and contains a block of data. Unit members will be discussed in more detail in section 4.2. The DATA BASE Structures descriptor provides details concerning all unit members. The ERRORS descriptor provides useful error diagnostics. Computer core requirement are under LDS (Local Dynamic Storage) and GDS (Global Dynamic Storage).

Example 3.1 depicts user documentation for the Atmospheric Module (ATM). Included in the documentation are various types of user parameters: integer (I), real single (RS), and alphanumeric (A). Two examples of table members are included. Example 3.1 will be referred to extensively in Section 4 with further examples demonstrating how to use the documentation.

#### Example 3.1 Atmospheric Module Prologue

```
***
*   PURPOSE - BUILD TABLE OF ATMOSPHERIC MODEL DATA AS FUNCTIONS
*               OF ALTITUDE
*
*   AUTHOR - SWP(L03/00/00)
*
*   INPUT
*   USER PARAMETERS
*       DELH      ALTITUDE INCREMENT FOR OUTPUT, M (FT)
*       H1        GROUND LEVEL ALTITUDE REFERENCED TO SEA LEVEL
*                   M (FT)
*       IUNITS    INPUT UNITS CODE
*                   =2HSI,  INPUTS ARE IN SI UNITS
*                   =7HENGLISH, INPUTS ARE IN ENGLISH UNITS
*       NHO       NUMBER OF ALTITUDES FOR OUTPUT ATMOSPHERIC
*                   FUNCTIONS
*       P1        ATMOSPHERIC PRESSURE AT GROUND LEVEL
*                   N/M**2 (LBF/FT**2)
*       IPRINT    PRINT CODE FOR FORTRAN WRITE
*                   0 NO PRINT DESIRED
*                   1 INPUT PARAMETER PRINT ONLY
*                   2 OUTPUT PRINT ONLY
*                   3 BOTH INPUT PARAMETER AND OUTPUT PRINT
```

```

*
* REAL USER PARAMETER LIMITS - SI UNITS
*
*   PARAMETER      MINIMUM      MAXIMUM      DEFAULT
*   DELH           1.0          100000.0      100.0
*   H1             -300.0        10000.0        0.0
*   P1             26200.6517    110000.0      101325.0
*
* REAL USER PARAMETER LIMITS - ENGLISH UNITS
*
*   PARAMETER      MINIMUM      MAXIMUM      DEFAULT
*   DELH           3.2808        328083.99     328.0839
*   H1             -984.2519      32808.399     0.0
*   P1             574.212        2297.3978     2116.2167
*
* INTEGER/LOGICAL/ALPHA PARAMETER LIMITS
*
*   PARAMETER      MINIMUM      MAXIMUM      DEFAULT
*   IPRINT          0            3            3
*   NHO             1            60            1
*
* MEMBER
*   ATM( IN )
*
* TEMPORARIES
*   MEMBER
*   SCRATCH( TAB1 )
*
* OUTPUT
* SYSTEM PARAMETER
*   NERR           EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED
*                   DURING EXECUTION OF A FUNCTIONAL MODULE. NERR
*                   SET TO .TRUE. IF ERROR ENCOUNTERED.
*
* MEMBER
*   ATM( TMOD )
*
* DATA BASE STRUCTURES
*   ATM( IN )      CONTAINS DATA INPUT TO ATM IN FOLLOWING FORMAT
*
*                   RECORD  FORMAT  DESCRIPTION
*                   -----
*   1   3RS        ALT, TEMP, RELATIVE HUMIDITY
*                   (ALTITUDE, "ALT", IS
*                   REFERENCED TO SEA LEVEL AND
*                   SHOULD NOT BE LESS THAN USER
*                   PARAMETER H1.)
*
*                   .
*
*                   ALTITUDE UNITS          M(FT)
*                   TEMPERATURE UNITS        KELVIN(RANKINE)
*                   RELATIVE HUMIDITY        PERCENT
*
* SCRATCH( TAB1 )
*   TEMPORARY TWO-DIMENSIONAL TYPE 1 DATA TABLE
*   INDEPENDENT VARIABLES
*     1. ALTITUDE
*     2. ORDERED POSITION
*   DEPENDENT VARIABLES IN FOLLOWING ORDER
*     TEMPERATURE
*     HUMIDITY
*
*   ATM( TMOD )    OUTPUT TWO-DIMENSIONAL TYPE 1 DATA TABLE OF

```



## 4. Control Statements

Described in this section are ten of the most frequently used statements for preparing a PAS module for execution. A complete description of all the ANOPP control statements can be found in reference 5.

Each executive control statement has a specific format indicated in the following subsections. All control statement formats adhere to the following conventions:

- \* Each control statement directive is a free-form sequence, using columns 1 to 80
- \* A control statement may begin in any column and continue across as many as 5 lines to complete the directive.
- \* Each control statement is terminated by the \$ character.
- \* Comments may appear in columns following the \$ character terminator.
- \* Comments may continue across lines only if the first character on the line is the \$ character terminator.

The general format of a control statement (CS) is as follows:

CSNAME OPERANDS \$ COMMENTS

CSNAME            control statement name  
Listed below are the twelve most frequently used  
ANOPP control statements:

ANOPP	STARTCS
LOAD	UNLOAD
PARAM	EVALUATE
EXECUTE	ENDCS
UPDATE	TABLE

OPERANDS        These are the operand fields that are required for each  
of the individual control statements.

COMMENTS       Any user desired comment can be included.

ANOPP control statements can be divided into two categories, Single Directive and Multiple Directive. As the name implies, single directive control statements require only one statement to execute a given function. These commands are described in Section 4.1. Multiple directive control statements, described in Section 4.2, require sub-commands to execute a given function.

## 4.1 Single Directive Control Statements

- 4.1.1 ANOPP      Purpose:      The ANOPP control statement is the first CS in the input deck.
- Format:      ANOPP JECHO=.TRUE. JLOG=.TRUE. \$
- JECHO:    print control during edit phase  
                                 JLOG:    print control during execution phase
- A complete list of system parameters has been tabulated on page 3-8 of the ANOPP User's Manual reference 5.
- 4.1.2 STARTCS    Purpose:      The STARTCS control statement is the second CS in the input deck. STARTCS begins the execution.
- Format:      STARTCS \$
- 4.1.3 LOAD        Purpose:      The LOAD control statement loads unit members from an ANOPP library which has been previously stored on an external file via the UNLOAD control statement.
- Format:      LOAD/external file/unit1,...,unitn \$
- Example:      Load the unit ATM from the external file LIBRARY. Unit ATM contains tables which are required by the PRT module.
- LOAD/LIBRARY/ ATM \$
- 4.1.4 UNLOAD     Purpose:      The UNLOAD control statement establishes an ANOPP library for storage of one or more units on an external file.
- Format:      UNLOAD/external file/unit1,...,unitn \$
- Example:      Create an ANOPP library with units UN1 and UN2 and store it on external file EXTFIL.
- UNLOAD/EXTFIL/UN1,UN2 \$
- 4.1.5 PARAM      Purpose:      The PARAM control statement establishes values of one or more user parameters.
- Format:      PARAM pname<sub>1</sub>=value<sub>1</sub>,...,pname<sub>n</sub>=value<sub>n</sub> \$

pname: user parameter name  
 value: any required integer, real single precision, logical, or alphanumeric value

Example: Referring to example 3.1, assign values to the following user parameters:

DELH	altitude increment for output	150. m
H1	ground level altitude	10. m
NHO	number of altitudes for output atmospheric functions	50
IPRINT	print option	output only
IUNITS	units	metric
PARAM	DELH=150.,H1=10.,NHO=50, IPRINT=2,IUNITS=2HSI \$	

4.1.6 EVALUATE Purpose: The EVALUATE control statement establishes the value of a user parameter via an arithmetic expression.

Format: EVALUATE Pname=expression \$

Pname: user parameter name  
 expression: a sequence of constants, user parameters and functions separated by operators and parentheses

The arithmetic operators are as follows:

+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation

It is important to note that the arithmetic operators '+' and '-' must be preceded and followed by at least one blank space when used in the EVALUATE statement.

Additional functions are available as shown in Table 1.

Example: Evaluate the nondimensional velocity V given a velocity of 102 meters per second and the default speed of sound, C, equals 340.294 meters per second.

EVALUATE V=102./ C \$

Name	Definition	Number of Arguments	Type of Arguments	Example
ABS	$ x $	1	any type	$Y = \text{ABS}(X)$
ANTILOG	$10^x$	1	I,RS,RD	$Y = \text{ANTILOG}(X)$
COS	$\cos(x)$	1	any type	$Y = \text{COS}(X)$ X in degrees
INT	convert to integer	1	any type	$Y = \text{INT}(X)$
LOG	$\log_{10}(x), x > 0$	1	I,RS,RD	$Y = \text{LOG}(X)$
REAL	convert to real	1	any type	$Y = \text{REAL}(X)$
SIN	$\sin(x)$	1	any type	$Y = \text{SIN}(X)$ X in degrees
SQRT	$\sqrt{x}, x \geq 0$	1	any type	$Y = \text{SQRT}(X)$
TAN	$\sin(x)/\cos(x)$	1	any type	$Y = \text{TAN}(X)$ X in degrees

Table 1. Generic Functions for the EVALUATE Control Statement

- 4.1.7 EXECUTE     Purpose:     The EXECUTE control statement calls a specific functional module into execution.
- Format:     EXECUTE functional module name \$
- Example:     Execute the Geometry module, GEO.
- EXECUTE GEO \$
- 
- 4.1.8 ENDCS     Purpose:     The ENDCS control statement is the last line in the input deck and terminates the ANOPP run.
- Format:     ENDCS \$



## 4.2 Multiple Directive Control Statements

The control statements discussed so far are single directive statements. The UPDATE and TABLE statements are multiple directive statements. The purpose of these two statements is to provide a unit of information to a module.

As indicated in figure 4, a library is a collection of units and a unit is a collection of members. Two types of members are described, data members and tables. Data members are input using the UPDATE control statement and provide a unit of information to a module that does not require interpolation. A unit requiring interpolation is input using the TABLE control statement. A table is a member with a specific structure.

4.2.1 UPDATE	<u>Purpose:</u>	The UPDATE control statement allows the user to input a unit.
	<u>Format:</u>	UPDATE NEWU=unitname SOURCE=* \$  unitname: name of data unit onto which new members are to be generated
-ADDR	<u>Purpose:</u>	The -ADDR control statement allows the user to input a member on a specific unit with the aid of the UPDATE control statement.
	<u>Format:</u>	-ADDR OLDM=* NEWM=mname FORMAT=format \$  mname: input member name  Valid format specifications are:  FORMAT=0 Unformatted FORMAT=2HCI Card Image FORMAT=nHet,...,et\$ Fixed Length Format FORMAT=nH*et,...,et\$ Variable Length Format  n: number of Hollerith characters in the format specification valid element types (et) are:  I Integer RS Real Single CS Complex Single L Logical A Alphanumeric  The input deck follows the -ADDR statement, is separated by blanks or commas, and may take as many lines as necessary.

**END\***      **Purpose:**      The END\* control statement signals the termination of input to the unit. This statement is also used with the TABLE and DATA statements.

**Example:**      A user is required to input unit member OBSERV(COORD) with each record having three real single precision values.

```
UPDATE NEWU=OBSERV SOURCE=* $
-ADDR OLDLM=* NEWM=COORD FORMAT=4H3RS$ $
      10.   20.   30.   $
      20.   20.   20.   $
      30.   20.   10.   $
END* $
```

**Example:**      A user is required to input unit SFIELD which consists of members FREQ, THETA, and PHI. This unit member represents the 1/3-octave band frequencies, polar directivity angles, and azimuthal directivity angles required by every source noise module for calculation purposes.

```
UPDATE NEWU=SFIELD SOURCE=* $
-ADDR OLDLM=* NEWM=FREQ FORMAT=4H*RS$ $
      50.   63.   80.  100.  125.
      160.  200.  250.  315.  400.
      500.  630.  800. 1000. 1250.
      1600. 2000. 2500. 3150. 4000.
      5000. 6300. 8000.10000.   $
ADDR OLDLM=* NEWM=THETA FORMAT=4H*RS$ $
      10.  30.  50.  70.  90. 110.
      150. 170. $
-ADDR OLDLM=* NEWM=PHI FORMAT=4H*RS$ $
      0. $
END* $
```

**Example:**      A user is required to input unit ATM which consists of the member IN. This unit member is required as input to the Atmosphere Module. It consists of a temperature and humidity profile as a function of altitude.

```
UPDATE NEWU=ATM SOURCE=* $
-ADDR OLDLM=* NEWM=IN FORMAT=4H3RS$ $
      0.   313.2  70.   $
      1000. 306.7  70.   $
      2000. 300.2  70.   $
      3000. 293.7  70.   $
      4000. 287.2  70.   $
      5000. 280.7  70.   $
END* $
```

#### 4.2.2 TABLE\*\*

Purpose: The TABLE control statement builds a table member in accordance with a set of user supplied instructions for interpolation.

Format: Type 1 Tables (only type currently available).

```
TABLE UNIT(MEMBER) 1 SOURCE=* $
  INT=0,1,2
  IND1=RS,n1,2,2, independent variable values
                    separated by commas or blanks
  IND2=RS,n2,2,2, independent variable values
                    separated by commas or blanks
  IND3=RS,n3,2,2, independent variable values
                    separated by commas or blanks
  IND4=RS,n4,2,2, independent variable values
                    separated by commas or blanks
  DEP=RS, dependent variable values separated by
                    commas or blanks
END* $
```

The integer values n1,...,n4 are the number of values of the corresponding independent variables. If the table has less than four dimensions, then fewer independent variables are needed. If the independent variable is ordered position, then the RS is replaced by a 0 and no independent variable values are needed. Independent and dependent variable values may take as many lines as needed.

Example 1: The following two functions, pressure and temperature, are input as table ATM(SAMPLE) using ordered position. The tabulated pressure values are entered first followed by the temperature values. IND2 is used to indicate ordered position by replacing RS with 0 and setting n2 equal to 2 indicating the two functions, pressure and temperature.

<u>altitude</u>	<u>pressure</u>	<u>temperature</u>
0.	2116.	510.
2000.	1965.	506.
4000.	1824.	502.
6000.	1692.	498.

```
TABLE ATM(SAMPLE) 1 SOURCE=* $
  INT=0 1 2
  IND1=RS 4 2 2 0. 2000. 4000.
                    6000.
  IND2=0 2 2 2
  DEP=RS 2116. 1965. 1824. 1692.
                    510. 506. 502. 498.
```

---

\*\* See Appendix B of this manual for a detailed discussion of the TABLE control statement

END\* \$

Example 2: The following example is a table of the pressure and the skin friction loadings as functions of the spanwise station (XI1), the chordwise station (XI2), and the in-plane station (PSI). This table is built by PLD or it can be built by the user if the loading information is available. In this table, beside the three independent variables XI1, XI2, and PSI, there are two ordered positions: the first one is the pressure loading and the second one is the skin friction loading.

XI1	XI2	PSI
0.7000	0.0000	0.000
0.8000	1.2566	
0.8500	1.8850	
0.9000	2.5133	
0.9500	3.1416	
0.9750	3.7699	
0.9970	4.3982	
	5.0265	
	5.6549	
	6.2832	

The first 70 numbers are the pressure loadings, and the next 70 numbers are the skin friction loadings. The table is formed as follows:

```
TABLE PLD (LOADS      ) 1 SOURCE=* $
INT=          0    1    2
IND1= RS      7    2    2
      0.7000  0.8000  0.8500  0.9000
      0.9500  0.9750  0.9970
IND2= RS     10    2    2
      0.0000  1.2566  1.8850  2.5133
              3.1416  3.7699  4.3982
              5.0265  5.6549  6.2832
IND3= RS      1    1    1
      0.0000
IND4= 0        2    0    0
DEP= RS
      0.0649  0.0718  0.0907  0.1129  0.1279
      0.0114  0.0126 -0.1025 -0.1491  0.1600
      -0.1681 -0.1873 -0.4114 -0.4462 -0.2604
      -0.2728 -0.3350 -0.4054 -0.4530 -0.0585
      -0.0724 -0.1310 -0.2345 -0.2339 -0.2190
      -0.2257 -0.8863 -1.0035  0.1932  0.2041
      0.2360  0.2774  0.3473  0.3634  0.4279
      0.0501  0.0775  0.0873  0.0948  0.0905
      0.1717  0.1965  0.0542  0.0726  0.0842
```

0.0959	0.1018	0.0654	0.0792	0.0351
0.0441	0.0517	0.0596	0.0635	0.0385
0.0462	0.0540	0.0674	0.0745	0.0814
0.0906	0.0860	0.0933	0.0698	0.0786
0.1000	0.1251	0.1416	-0.0249	-0.0258
0.0009	0.0011	0.0013	0.0014	0.0016
0.0017	0.0023	0.0010	0.0012	0.0014
0.0015	0.0017	0.0018	0.0025	0.0011
0.0014	0.0016	0.0017	0.0019	0.0021
0.0028	0.0014	0.0018	0.0020	0.0023
0.0025	0.0027	0.0037	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0014	0.0018	0.0020	0.0022	0.0025
0.0027	0.0036	0.0011	0.0014	0.0016
0.0017	0.0019	0.0021	0.0028	0.0010
0.0012	0.0014	0.0015	0.0017	0.0018
0.0025	0.0009	0.0012	0.0013	0.0014
0.0016	0.0017	0.0023	0.0009	0.0011
0.0013	0.0014	0.0016	0.0017	0.0023

END\* \$

## 5. Example Programs

In this section, examples will be given showing how to obtain user documentation for the ATM module and prepare input for execution. The examples include the control statements necessary to prepare any module for execution.

### Example 5.1

To obtain the user documentation for the ATM module, the following ANOPP input deck can be executed. Appendix A lists the names of modules currently included in PAS-ANOPP. To obtain user documentation for any one of these modules, replace ATM in the following example with the name of the desired module.

```
ANOPP JECHO=.TRUE. $
STARTCS $
LOAD /LIBRARY/ MANUAL $
MEMLIST MANUAL(ATM) FORMAT=2HCI $
ENDCS $
```

The MEMLIST is an ANOPP control statement which allows a user to list the contents of a unit member. The unit MANUAL contains documentation for all functional modules. The member ATM contains documentation for the ATM module.

### Example 5.2

A demonstration of the use of the Atmospheric Module (ATM) is presented in this example. The purpose of this module is to generate tables of atmospheric data that can be used by other modules for subsequent calculations. One table is generated in this example. This table provides conditions for a standard sea level atmosphere based on a 70% relative humidity (i.e. 0.2 percent mole fraction). Refer to the Atmospheric Module prologue, presented as Example 3.1, for more information concerning the input and output of this module.

```
ANOPP JECHO=.TRUE. $
STARTCS $
$
$ create the required input data base members
$
UPDATE NEWU=ATM SOURCE=* $
  -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
      0.      288.15    70.    $
      200.    286.85    70.    $
      400.    285.55    70.    $
```

	600.	284.25	70.	\$
	800.	282.95	70.	\$
	1000.	281.65	70.	\$

```

END* $
$
$ generate atmospheric properties
$
PARAM DELH=100. H1=0. NH0=11 P1=101325. IPRINT=3 $
$
EXECUTE ATM $
$
ENDCS $

```

### Example 5.3

The geometry module (GEO) is executed in this example. For any module to function properly, it must be supplied with certain tables or units of information. Normally the data can be generated by one module and then used in subsequent modules. In some cases, it may be more convenient for the user to provide input data required by a module. This is accomplished using the UPDATE control statement. For example, when examining pages 4-5 and 4-6 of the ANOPP User's Manual (ref. 5), it can be seen that the Geometry Module, GEO, requires the following data base structures: ATM(TMOD), FLI(PATH), and OBSERV(COORD) as input. The table ATM(TMOD) will be generated using the Atmospheric Module, ATM. The unit member FLI(PATH) can be generated by the SFO modules or it can be generated by the user. A detailed description of the unit member FLI(PATH) is given on page 4-7 of reference 6.

```

ANOPP JECHO=.TRUE. JLOG=.TRUE. $
STARTCS $
$
$ demonstration problems for geometry module
$
$ create required input data base members
$
UPDATE NEWU=ATM SOURCE=* $
  -ADDR OLDLM=* NEWLM=IN FORMAT=4H3RS$ $
    0.      536.670  50.      $
    200.    535.957  50.      $
    400.    535.244  50.      $
    600.    534.530  50.      $
    800.    533.817  50.      $
    1000.   533.104  50.      $
    1500.   532.604  50.      $
    2000.   532.236  50.      $
    2500.   532.082  50.      $
END* $
$
PARAM      DELH=100.   H1=0.      UNITS=7HENGLISH
           NH0=26     P1=2116.22  IPRINT=3  $
$

```

```

EXECUTE ATM $
$
$
UPDATE NEWU=FLI SOURCE=* $
    -ADDR OLDLM=* NEWM=PATH FORMAT=5H10RS$ $
        0.0      0.    50.   -1000.    0.  0.    0.  0.    0.  0.    $
        20.0     700.   50.   -1000.    0.  0.    0.  0.    0.  0.    $
        40.0    1400.   50.   -1000.    0.  0.    0.  0.    0.  0.    $
        60.0    2100.   50.   -1000.    0.  0.    0.  0.    0.  0.    $
        80.0    2800.   50.   -1000.    0.  0.    0.  0.    0.  0.    $
END* $
$
UPDATE NEWU=OBSERV SOURCE=* $
    -ADDR OLDLM=* NEWM=COORD FORMAT=4H3RS$ $
        100.      50.      5.      $
        100.      0.      10.      $
        1000.     -50.      5.      $
        1000.      0.      10.      $
        2000.     100.      5.      $
        2000.    -100.     10.      $
END* $
$
$    level flight path and START=10 and STOP=50
$
PARAM CTK=1.0  START=10.  STOP=50. $
$
EXECUTE GEO $
$
$
ENDCS $

```



## 6. Module Update

### 6.1 Propeller Performance (PRP) Module Modification

An error was found in equation (36) of the Aircraft Noise Prediction Program Theoretical Manual, Propeller Aerodynamics and Noise (ref. 3), page 10.5-9. This equation computes the resultant velocity of the fluid in the disk plane in the direction of rotation. Originally the equation was

$$V_{\psi}(r,\psi) = -[ (r + \lambda \sin \alpha_p \cos \psi) (1 - a_2) ]$$

The correct equation becomes

$$V_{\psi}(r,\psi) = -[ (r + \lambda \sin \alpha_p \sin \psi) (1 - a_2) ]$$

where	$r$	spanwise stations, re $R$
	$\lambda$	local advance ratio
	$\alpha_p$	propeller angle-of-attack, rad.
	$\psi$	blade rotation angle, rad.
	$a_2$	induced tangential velocity component, re $rR\Omega$
	$R$	blade length
	$\Omega$	angular velocity of blade, rad/s

Figure 5 shows results from the incorrect equation and the modified equation for non-zero angle-of-attack. For a zero angle-of-attack simulation no error is involved in the prediction.

## 6.2 Subsonic Propeller Noise (SPN) Module Modification

The PAS modules have been continuously updated and validated by comparing with measured data. A major modification was made for inclusion of shaft angle-of-attack in the Subsonic Propeller Noise (SPN) module.

### NOMENCLATURE

$c_o$	ambient speed of sound
$f$	function defining blade surface
$\ell$	local force per unit area of blade acting on fluid
$\ell_r$	component of loading vector in direction of radiation vector, $(\ell_r = \ell_i \hat{r}_i)$
$M$	source Mach number
$M_r$	component of source Mach vector in direction of radiation vector, $(M_r = M_i \hat{r}_i)$
$\mathbf{n}$	blade surface normal vector
$p'$	acoustic pressure
$p'_L$	acoustic pressure produced by loading
$p'_T$	acoustic pressure produced by thickness
$\mathbf{r}$	distance from source point at emission time to observer
$\hat{\mathbf{r}}$	unit vector in direction $\mathbf{r}$
$S$	surface area
$t$	time at which noise signal is received by observer
$\mathbf{v}$	source velocity vector
$V_F$	forward velocity of aircraft
$v_n$	source velocity component in direction of blade normal, $(v_n = v_i n_i)$
$\mathbf{x}$	observer position in ground fixed frame
$\mathbf{y}$	source position in ground fixed frame
$\alpha$	aircraft angle-of-attack
$\eta$	source position in blade fixed frame
$\tau$	time at which noise signal is emitted at source position
$\psi$	angle between $x_1$ and $\eta_1$ axes, $(\psi = \Omega \tau)$
$\Omega$	angular velocity of blade

## SPN Noise Model

Originally, the PAS ANOPP noise module SPN did not include propeller angle-of-attack (inflow angle) in the module formulation. Modifications were made to SPN to incorporate the effects due to angle-of-attack and the new version was tested and compared with DNW data. In the following discussion, the analysis pertains to the Full Blade Formulation. The physical model on which the module is based expresses the acoustic pressure as (ref. 3).

$$4\pi p'_L(\mathbf{x}, t) = \frac{1}{c_o} \int_{f=0} \left[ \frac{\dot{\ell}_i \hat{\mathbf{r}}_i}{r(1-M_r)^2} \right]_{\tau} dS + \int_{f=0} \left[ \frac{\ell_r - \ell_i M_i}{r^2(1-M_r)^2} \right]_{\tau} dS$$

(6.1a)

$$+ \frac{1}{c_o} \int_{f=0} \left[ \frac{\ell_r (r\dot{M}_i \hat{\mathbf{r}}_i + c_o M_r - c_o M^2)}{r^2(1-M_r)^3} \right]_{\tau} dS$$

$$4\pi p'_T(\mathbf{x}, t) = \int_{f=0} \left[ \frac{\rho_o v_n (r\dot{M}_i \hat{\mathbf{r}}_i + c_o M_r - c_o M^2)}{r^2(1-M_r)^3} \right]_{\tau} dS$$

(6.1b)

where

$$p'(\mathbf{x}, t) = p'_L(\mathbf{x}, t) + p'_T(\mathbf{x}, t)$$

(6.1c)

Since the integrands depend on vector operations, appropriate reference frames must be established. Three reference frames, which are illustrated in figure 6, are employed in the computational scheme. These frames are the ground (medium) fixed  $\mathbf{x}$ -frame, the aircraft fixed  $\mathbf{X}$ -frame and the blade fixed  $\eta$ -frame. At the initial time, the propeller hub is located at the origin of the  $\mathbf{x}$ -frame. The  $x_3$  axis defines the flight direction. Initially, the  $\mathbf{X}$ -frame coincides with the  $\mathbf{x}$ -frame but afterwards translate at the constant rate  $V_F$ , the aircraft forward velocity. Operations involving blade normals or surface pressures are more easily computed in the  $\eta$ -frame. But it is more convenient to express the source position,  $\mathbf{y}$ , in the  $\mathbf{x}$ -frame then compute  $\mathbf{r}=\mathbf{x}-\mathbf{y}$  and  $\hat{\mathbf{r}}=\mathbf{r}/r$  in the  $\mathbf{x}$ -frame and transform the vector components to the  $\eta$ -frame. Considering equation (6.1), it is seen that the quantities that need to be revised to include angle-of-attack are  $\mathbf{r}$ ,  $\hat{\mathbf{r}}$ ,  $v_n$  and  $\mathbf{M}$ . Note that  $v_n$  and  $\mathbf{M}$  are

based on source absolute velocity. No correction is needed for the source absolute acceleration,  $\dot{\mathbf{M}}$ , since the aircraft forward velocity,  $V_F$ , is constant. Also, the retarded time equation (RTE), which must be solved to establish emission time ( $\tau$ ) for each observer time ( $t$ ), must be modified for angle-of-attack.

The observer and source locations in the  $\mathbf{x}$ -frame are, respectively

$$\mathbf{x} = \mathbf{V}_F t + \mathbf{x}_o \quad (6.2)$$

$$\mathbf{y} = \mathbf{V}_F \tau + \mathbf{T}_\alpha \mathbf{T}_\psi \boldsymbol{\eta} \quad (6.3)$$

where

$$\mathbf{T}_\psi = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

and

$$\mathbf{T}_\alpha = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad (6.5)$$

Thus, the matrix form of equation (6.3) is

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \cos\alpha \cos\psi & -\cos\alpha \sin\psi & \sin\alpha \\ \sin\psi & \cos\psi & 0 \\ -\sin\alpha \cos\psi & \sin\alpha \sin\psi & \cos\alpha \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ V_F \tau \end{bmatrix} \quad (6.6)$$

Equation (6.6) will be called the first correction for  $\alpha$  and was implemented in the module software. For  $\alpha=0$  the original component equations are obtained. The above revision for  $\alpha$  allows the radiation vector, given by

$$\mathbf{r} = \mathbf{x} - \mathbf{y} \quad (6.7)$$

to be computed in the  $\mathbf{x}$ -frame. But  $\mathbf{r}$  must be transformed to the  $\boldsymbol{\eta}$  -frame for the calculation of  $p'$  according to equation (6.1). This transformation is

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}_\eta = T_\Psi^{-1} T_\alpha^{-1} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}_x \quad (6.8)$$

where

$$T_\Psi^{-1} T_\alpha^{-1} = \begin{bmatrix} \cos \psi \cos \alpha & \sin \psi & -\cos \psi \sin \alpha \\ -\sin \psi \cos \alpha & \cos \psi & \sin \psi \sin \alpha \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (6.9)$$

Equation (6.8) represents the second correction for  $\alpha$  in the SPN module. The absolute velocity for each source point is expressed as

$$\mathbf{v} = \mathbf{V}_F + \boldsymbol{\Omega} \times \boldsymbol{\eta} \quad (6.10)$$

In the  $\eta$  -frame, equation (6.10) can be written as

$$\mathbf{v} = T_\Psi^{-1} T_\alpha^{-1} \begin{bmatrix} 0 \\ 0 \\ V_F \end{bmatrix} + \boldsymbol{\Omega} \times \boldsymbol{\eta} \quad (6.11)$$

From equation (6.11), it is found that the components of  $\mathbf{v}$  are

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -\Omega \eta_2 - V_F \sin \alpha \cos \psi \\ \Omega \eta_1 + V_F \sin \alpha \sin \psi \\ V_F \cos \alpha \end{bmatrix} \quad (6.12)$$

This is the third correction for  $\alpha$ . As indicated in equation (6.1), emission times must be found for the RTE:

$$|\mathbf{x}(t) - \mathbf{y}(\tau)|^2 = c_o^2 (t - \tau)^2 \quad (6.13)$$

Using equations (6.2) and (6.3) allows the above relation to be stated in terms of  $\eta$  -frame components as

$$\left| T_{\Psi}^{-1} T_{\alpha}^{-1} [x_o + V_F(t - \tau)] - \eta \right|^2 = c_o^2(t - \tau)^2 \quad (6.14)$$

where

$$T_{\Psi}^{-1} T_{\alpha}^{-1} [x_o + V_F(t - \tau)] = T_{\Psi}^{-1} T_{\alpha}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 + V_F(t - \tau) \end{bmatrix} \quad (6.15)$$

Equation (6.14) produces the following retarded time relation

$$A \phi^2 + B \phi + C + \cos(\phi + D) + E \phi \cos(\phi + F) = 0 \quad (6.16)$$

where

$$\phi = \Omega(\tau - t)$$

$$A = \frac{c_o^2 - V_F^2}{2\eta x^* \Omega^2}$$

$$B = \frac{V_F[-x_1^* \sin \alpha + (x_3^* - \eta_3) \cos \alpha]}{\Omega \eta x^*}$$

$$C = -\frac{[x^{*2} + \eta^2 + (x_3^* + \eta)^2]}{2x^* \eta}$$

$$D = \psi_{\eta} - \psi_x + \Omega t$$

$$E = \frac{V_F \sin \alpha}{\Omega x^*}$$

$$F = -\psi_{\eta} + \Omega t$$

$$x_1^* = x_1 \cos \alpha - x_3 \sin \alpha$$

$$x_3^* = x_3 \cos \alpha + x_1 \sin \alpha$$

$$x^* = \sqrt{x_1^{*2} + x_2^2}$$

$$\eta = \sqrt{\eta_1^2 + \eta_2^2}$$

$$\psi_x = \tan^{-1}\left(\frac{x_2}{x_1}\right)$$

$$\psi_\eta = \tan^{-1}\left(\frac{\eta_2}{\eta_1}\right)$$

This represents the fourth correction for  $\alpha$ . In addition to the Full Blade Formulation, there are three approximate options in the Subsonic Propeller Noise module. Corrections for  $\alpha$  are also included in the mean-surface, compact chord, and point source approximations.

Originally, the SPN iteration procedure had a number of checks, which are approximations, for the initial guess in Newton's method. If these checks were not satisfied the program stops and an error message results indicating the TPN module is more appropriate. This happened for some "non-severe" cases (low RPM, low helical Mach no.). With the above described coding, no attempt was even made in the Newton iteration scheme. The code has now been modified to always attempt the iteration. This change resulted in the previous problematical cases producing plausible sound levels. The TPN procedure should never be used for a subsonic propeller.

## 7. Description of Prediction Capabilities

ANOPP PAS has the capability of predicting wind tunnel and flyover noise. PAS noise prediction requires knowledge of the propeller geometry, propeller operating state, source to observer geometry, and atmospheric data as shown in Table 2.

From the propeller geometry, the Rotating Blade Shape (RBS or IBS) module generates a functional representation of the blade surface suitable for aerodynamic and aeroacoustic calculations. Subsequently, pressure and blade section lift distributions are computed by the Rotating Blade Aerodynamic (RBA or IBA) module, then blade skin friction and section drag distributions are computed by the Boundary Layer (BLM or IBL) module.

There are two options in the Propeller Performance (PRP) module. The first option is to match the computed power coefficient with the measured power coefficient. An initial guess of the blade 3/4 radius pitch angle is required for the input. The computed power coefficient is compared to the measured value. Iteration is performed using the secant method until the computed and measured power coefficient converge. Thus, the absorbed power for the predictions match the measured data, but the blade 3/4 radius pitch angles most likely will differ. The other option is to input the correct 3/4 radius pitch angle and PRP is executed only one time to compute the absorbed power coefficient. The final blade pressure and skin friction distributions are determined using the Propeller Loads (PLD) module.

From the blade geometry and performance data, the propeller noise signature is predicted by the Subsonic Propeller Noise (SPN) module. This module produces acoustic time histories and narrow band spectra of loading, thickness, and total noise. There are two options to use SPN. The first option is the noise prediction in a wind tunnel configuration and the second option is the creation of a noise bubble for further calculation for a flyover noise prediction. For the wind tunnel noise prediction, microphone (observer) locations are input in rectangular coordinates relative to the propeller hub. For the noise bubble (flyover prediction) observers are set in the polar and azimuthal directions with a chosen radius for the noise bubble. The default radius is a distance of five times the propeller radius. There are four methods which are available in SPN for computing the radiated acoustic field. They are the full blade surface formulation, mean surface approximation, compact chord approximation, and point source approximation. Users can choose one of the four methods depending on the computer execution time and the required precision of the prediction. Further comparisons of the results of the four methods will be discussed in Section 9. If the flight Mach number is greater than 0.7, then the Transonic



Propeller Noise (TPN) module should be used. The Propeller Trailing Edge Noise (PTE) module computes broadband noise from the propeller trailing edge.

For flyover predictions, additional calculations are required. Atmospheric properties are computed from the Atmospheric (ATM) and Absorption (ABS) modules. The Steady Flyover (SFO) module defines the aircraft flight path and the Geometry (GEO) module computes the range and directivity angles from observer to source at sound emission. The Tone Propagation (PRT) module propagates narrow band spectra from the source to the observer applying Doppler shift, spherical spreading, characteristic impedance, atmospheric absorption, and ground effect corrections. The Broadband Propagation (PRO) propagates the broadband spectra of PTE. The Noise Levels (LEV) module sums the noise if requested and computes OASPL and  $L_A$ . Finally, the Effective Perceive Noise Level (EFF) module computes EPNL.

Table 2. Input data requirements

Propeller Geometry

Airfoil Section Coordinates  
Chord Distribution  
Twist Distribution  
Leading Edge Displacement Distribution  
Blade Length  
Number of Blades

Propeller Operating State

Propeller RPM  
Forward Speed  
Absorbed Power  
Root Pitch Angle  
Nacelle Tilt Angle (angle-of-attack)

Source to Observer Geometry

Flight Path Angle  
Observer Positions

Atmospheric Data

Ambient Temperature Profile  
Ground Level Pressure

## 8. PAS Program Templates

This section contains ten templates which have been developed to demonstrate the types of problems that can be solved using ANOPP-PAS. Templates one and two demonstrate how the blade geometry is input using the improved and the original ANOPP-PAS modules. Templates three and four demonstrate how to compute the propeller performance. In template three, the blade pitch is known and the performance is computed directly by the PLD module. In template four, the pitch is unknown and an iterative scheme is used. Template five demonstrates how the propeller loads are calculated using the PLD module. Template six demonstrates how measured propeller loads can be input directly, bypassing the PLD module. Templates seven and eight demonstrate how the propeller noise is calculated using the SPN module. Template seven calculates the near-field noise. Template eight calculates the noise on a sphere around the propeller (i.e. "sound bubble") for propagation to the far-field. Finally, template nine demonstrates a simple flyover prediction and then template ten demonstrates how ANOPP-PAS can be used to add noise from multiple rotors such as the tilt rotor. Each template builds upon the information of the preceding template. The input and output of each module can be found in reference 3. In most cases this information is also available on line using the "man" command on UNIX systems and the HELP command on VMS systems. The ANOPP control statements are described in section 4 of this document. Additional information concerning the control statements can be found in reference 5.

### 8.1 Blade Geometry

#### 8.1.1 Template 1 - The Improved Version of PAS

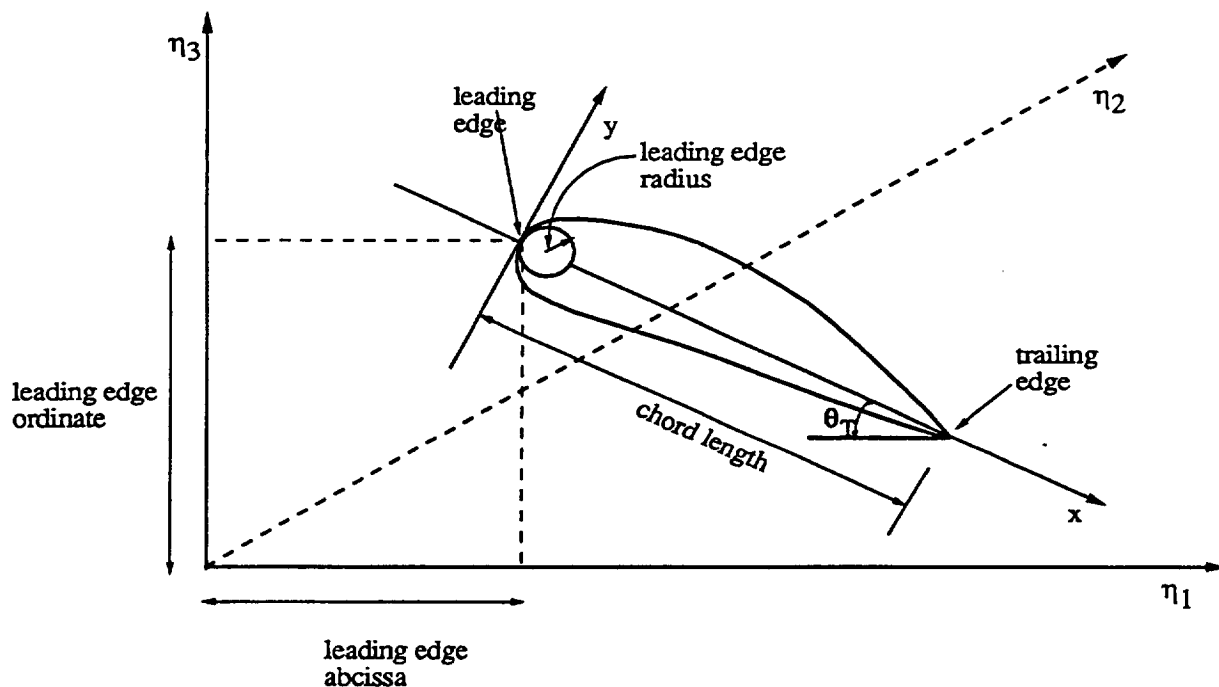
**Problem:** Given a propeller blade geometry with 5 identical cross sections, tables of cross sectional lift, drag and pressure coefficients are built in the given ranges of angle-of-attack and Mach number.

**Solution:** The first step is to transform the airfoil section data from Cartesian coordinate to the elliptical coordinate defined by the inverse Joukowski transformation. This procedure is performed in IBS. The second step is to compute the sectional lift coefficient using the Kutta-Joukowski theorem. Also, the pressure coefficient is computed using Bernoulli's equation. The compressibility correction is extended to subsonic flow by Karman-Tsien or Glauert compressibility corrections in IBA. Finally, the profile drag coefficient is computed by the method of Squire and Young in IBL.

Input the blade geometry as required in the Improved Blade Shape (IBS) Module. A description of the blade geometry can be found in reference 3. The blade cross section is described in rectangular coordinates. Chordwise locations are designated by  $x$  where  $x=0.0$  is the leading edge and equals  $x=1.0$  is the trailing edge. The upper surface  $y$  is input before the lower surface  $y$ . Coordinates  $x$  and  $y$  are normalized by cross section chord,  $c$ . In this template, there are 5 propeller cross sections with the same spatial coordinates. The improved modules are used to shorten the input and provides additional compressibility correction options. Beside the Cartesian coordinates of the cross sections, other informations about the propeller are required. After showing how many cross sections are given, the next five lines which have eight numbers in each are

- Spanwise station normalized by the blade radius  $R$
- Leading-edge abscissa as shown in the following plot normalized by  $R$
- Leading-edge ordinate as shown in the following plot normalized by  $R$
- Chord length, normalized by  $R$
- Leading-edge radius, re chord length of the cross section
- Blade twist angle measured positive clockwise looking from hub toward propeller tip
- Number of  $x,y$  pairs for the upper surface
- Number of  $x,y$  pairs for the lower surface

An illustration of the blade geometry is shown below.



The followings are considerations that users should remember to avoid errors and also to obtain better predictions.

- The spanwise stations (XI1), re R (blade length) array should be in the range of XI1 in the given blade geometry.

- The chordwise stations (XI2), re  $2\pi$ , are from 0.0 to 1.0. From 0.0 to 0.5, these are points in chordwise direction from trailing edge to leading edge for upper surface. For lower surface, XI2 is from 0.5 to 1.0 from leading edge to trailing edge. For more accurate results, it is important to refine the grids at the leading edge.

- Blade section angles-of-attack and Mach numbers should be input to adequately cover the range of the flight condition of the prediction.

```

ANOPP JECHO=.TRUE. JLOG=.FALSE. $
STARTCS $
$
PARAM      R      = 13.205          $ blade radius in ft
PARAM      IUNITS = 7HENGLISH       $ use English units
$
UPDATE NEWU=GEOM SOURCE=* $
  -ADDR OLDM=* NEWM=BLADE FORMAT=0 $
    5 $ five spanwise stations
    5 $ five identical airfoil sections
0.00 -0.0106  0.000  0.043  0.025  0.00  20  19 $ 0% station
0.25 -0.0106 -0.000  0.043  0.025 -2.25  20  19 $ 25% station
0.50 -0.0106 -0.001  0.043  0.025 -4.50  20  19 $ 50% station
0.75 -0.0106 -0.001  0.043  0.025 -6.75  20  19 $ 75% station
1.00 -0.0106 -0.002  0.043  0.025 -9.00  20  19 $100% station

1.00000      .00158      $
.97000      .00674      $
.94000      .01172      $
.85000      .02566      $
.79000      .03417      $
.73000      .04207      $
.67000      .04937      $
.61000      .05600      $
.55000      .06191      $
.49000      .06695      $
.43000      .07097      $
.37000      .07376      $
.31000      .07499      $
.25000      .07427      $
.19000      .07095      $
.13000      .06399      $
.07000      .05108      $
.01750      .02772      $
.00250      .01090      $
.00000      .00000      $
.00250      -.01090      $
.01000      -.02130      $
.04000      -.04035      $

```

.10000	-.05853	\$
.16000	-.06802	\$
.22000	-.07298	\$
.28000	-.07491	\$
.34000	-.07459	\$
.40000	-.07254	\$
.46000	-.06910	\$
.52000	-.06454	\$
.58000	-.05905	\$
.64000	-.05277	\$
.70000	-.04580	\$
.76000	-.03819	\$
.82000	-.02999	\$
.88000	-.02117	\$
.94000	-.01172	\$
1.00000	-.00158	\$

END\* \$

\$

\$

UPDATE NEWU=GRID SOURCE=\* \$

-ADDR OLDLM=\* NEWM=XI1 FORMAT=4H\*RS\$ \$

0.200	0.400	0.600	0.700	0.750	0.800
0.850	0.900	0.925	0.950	0.975	1.000 \$

-ADDR OLDLM=\* NEWM=XI2 FORMAT=4H\*RS\$ \$

0.00	0.025	0.05	0.10	0.15
0.20	0.25	0.30	0.35	0.40
0.41	0.42	0.43	0.44	0.45
0.46	0.47	0.48	0.49	0.50
0.51	0.52	0.53	0.54	0.55
0.56	0.57	0.58	0.59	0.60
0.625	0.65	0.70	0.75	0.80
0.85	0.90	0.95	0.975	1.00 \$

END\* \$

\$

\$

\$ The values of section Mach number and angle-of-attack in degrees

\$ at which the blade section aerodynamics are computed are given here:

\$

\$

UPDATE NEWU=IBA SOURCE=\* \$

-ADDR OLDLM=\* NEWM=MACH FORMAT=4H\*RS\$ \$

0.1	0.3	0.5	0.7	\$
-----	-----	-----	-----	----

-ADDR OLDLM=\* NEWM=ALPHA FORMAT=4H\*RS\$ \$

-6.0	-3.0	0.0	3.0	6.0 \$
------	------	-----	-----	--------

END\* \$

\$

\$ Choose the compressibility correction options

\$

EXECUTE IBS \$

\$

PARAM ICL = 1 \$ Glauert compressibility correction for the lift

\$ coefficient

PARAM ICP = 1 \$ Glauert compressibility correction for the pressure

\$ coefficient

\$

EXECUTE IBA \$

EXECUTE IBL \$

\$

```

$
UNLOAD /BLDGEOM/ IBS, IBA, IBL $
ENDCS $

```

### 8.1.2 Template 2 - Blade description using the original PAS modules.

**Problem:** Given a propeller blade geometry with 8 different cross sections, build tables of cross sectional lift, drag and pressure coefficients in the given ranges of angle-of-attack and Mach number.

**Solution:** This template similar to template 1 except it serves as example for the use of the original RBS, RBA and BLM modules. If IBS, IBA and IBL are preferred then in the blade geometry input, after the "8 \$" line, the next added line is 1, 1, 1, 1, 1, 1, 1, 1 \$ to show that there are 8 different cross sections. Note that the leading edge is input first. The same results are provided as in template 1 except the unit-members or table names should start with RBS, RBA or BLM instead of IBS, IBA and IBL.

```

ANOPP $
$
STARTCS $
$
$
$          specify 21 evenly spaced chordwise grid points
$
$
UPDATE NEWU=GRID SOURCE=* $
-ADDR OLDLM=* NEWM=XI2 FORMAT=4H*RS$ MNR=1 $
.00 .05 .10 .15 .20 .25 .30 .35
.40 .45 .50 .55 .60 .65 .70 .75
.80 .85 .90 .95 1.0 $
END* $
$
$
PARAM R=1.0          IUNITS=2H$ I $
$
$
CREATE GEOM$
UPDATE NEWU=GEOM SOURCE=* $
-ADDR NEWM=BLADE OLDLM=* FORMAT=0 $
8 $          NO. OF RADIAL SECTIONS
0.30000 -0.05690 .03090 .14375 .02166 28.5 15 14 $ sta 12
0.          0.0151 $
0.05        0.0475 $
0.1         0.0632 $

```

0.2	0.0846	\$																	
0.3	0.0950	\$																	
0.4	0.0979	\$																	
0.5	0.0950	\$																	
0.6	0.0846	\$																	
0.7	0.0718	\$																	
0.8	0.0544	\$																	
0.9	0.0324	\$																	
0.925	0.0151	\$																	
0.95	0.0116	\$																	
0.975	0.0069	\$																	
1.0	0.0000	\$																	
0.05	-0.0116	\$																	
0.1	-0.0185	\$																	
0.2	-0.0278	\$																	
0.3	-0.0324	\$																	
0.4	-0.0336	\$																	
0.5	-0.0324	\$																	
0.6	-0.0301	\$																	
0.7	-0.0266	\$																	
0.8	-0.0220	\$																	
0.9	-0.0168	\$																	
0.925	-0.0151	\$																	
0.95	-0.0116	\$																	
0.97	-0.0069	\$																	
1.0	0.0000	\$																	
0.45000	-.06932	.02730	.16300	.00919	21.5	15	14	\$	sta	18									
0.	0.0230	\$																	
0.05	0.0466	\$																	
0.1	0.0578	\$																	
0.2	0.0702	\$																	
0.3	0.0755	\$																	
0.4	0.0741	\$																	
0.5	0.0689	\$																	
0.6	0.0603	\$																	
0.7	0.0490	\$																	
0.8	0.0364	\$																	
0.9	0.0231	\$																	
0.925	0.0153	\$																	
0.95	0.0082	\$																	
0.975	0.0031	\$																	
1.0	0.0000	\$																	
0.05	0.0082	\$																	
0.1	0.0051	\$																	
0.2	0.0021	\$																	
0.3	0.	\$																	
0.4	0.	\$																	
0.5	0.	\$																	
0.6	0.	\$																	
0.7	0.	\$																	
0.8	0.	\$																	
0.9	0.	\$																	
0.925	0.	\$																	
0.95	0.	\$																	
0.975	0.	\$																	
1.0	0.	\$																	
0.60000	-.07300	.02450	.16875	.00395	16.9	15	14	\$	sta	24									
0.	0.0089	\$																	

```
0.75000  -.07450  .02050  .16575  .00352  13.5  15  14  $ sta 30
```



0.90000	-.06550	.01875	.14575	.00286	11.1	15	14	\$ sta 36
0.		0.0027	\$					
0.05		0.0189	\$					
0.1		0.0251	\$					
0.2		0.0343	\$					
0.3		0.0388	\$					
0.4		0.0388	\$					
0.5		0.0365	\$					
0.6		0.0331	\$					
0.7		0.0280	\$					
0.8		0.0217	\$					
0.9		0.0143	\$					
0.925		0.0103	\$					
0.95		0.0046	\$					
0.975		0.0023	\$					
1.0		0.	\$					
0.05		0.	\$					
0.1		0.	\$					
0.2		0.	\$					
0.3		0.	\$					
0.4		0.	\$					
0.5		0.	\$					
0.6		0.	\$					
0.7		0.	\$					
0.8		0.	\$					
0.9		0.	\$					
0.925		0.	\$					
0.95		0.	\$					
0.975		0.	\$					
1.0		0.	\$					
0.95000	-.06050	.01725	.13438	.00286	10.45	15	14	\$ sta 38
0.		0.0027	\$					
0.05		0.0189	\$					
0.1		0.0251	\$					
0.2		0.0343	\$					
0.3		0.0388	\$					
0.4		0.0388	\$					
0.5		0.0366	\$					
0.6		0.0331	\$					
0.7		0.0280	\$					
0.8		0.0217	\$					
0.9		0.0143	\$					
0.925		0.0103	\$					
0.95		0.0046	\$					
0.975		0.0023	\$					
1.0		0.	\$					
0.05		0.	\$					
0.1		0.	\$					
0.2		0.	\$					
0.3		0.	\$					
0.4		0.	\$					
0.5		0.	\$					
0.6		0.	\$					
0.7		0.	\$					
0.8		0.	\$					
0.9		0.	\$					
0.925		0.	\$					
0.95		0.	\$					

0.975	0.	\$											
1.0	0.	\$											
0.97500	-.05250	.01750	.11250	.00296	10.12	15	14	\$	sta	39			
0.	0.0020	\$											
0.05	0.0178	\$											
0.1	0.0244	\$											
0.2	0.0311	\$											
0.3	0.0340	\$											
0.4	0.0355	\$											
0.5	0.0340	\$											
0.6	0.0304	\$											
0.7	0.0259	\$											
0.8	0.0200	\$											
0.9	0.0133	\$											
0.925	0.0074	\$											
0.95	0.0029	\$											
0.975	0.0015	\$											
1.0	0.	\$											
0.05	0.	\$											
0.1	0.	\$											
0.2	0.	\$											
0.3	0.	\$											
0.4	0.	\$											
0.5	0.	\$											
0.6	0.	\$											
0.7	0.	\$											
0.8	0.	\$											
0.9	0.	\$											
0.925	0.	\$											
0.95	0.	\$											
0.975	0.	\$											
1.0	0.	\$											
0.99750	-.0125	.00925	.02950	.00296	9.83	15	14	\$	sta	39.9			
0.	0.002	\$											
0.05	0.0178	\$											
0.1	0.0244	\$											
0.2	0.0311	\$											
0.3	0.0340	\$											
0.4	0.0355	\$											
0.5	0.0340	\$											
0.6	0.0304	\$											
0.7	0.0259	\$											
0.8	0.0200	\$											
0.9	0.0133	\$											
0.925	0.0074	\$											
0.95	0.0029	\$											
0.975	0.0015	\$											
1.0	0.	\$											
0.05	0.	\$											
0.1	0.	\$											
0.2	0.	\$											
0.3	0.	\$											
0.4	0.	\$											
0.5	0.	\$											
0.6	0.	\$											
0.7	0.	\$											
0.8	0.	\$											
0.9	0.	\$											

```

0.925      0.      $
0.95       0.      $
0.975      0.      $
1.0        0.      $
END*$

$
$
$ compute smooth blade shape using RBS module
$
$
EXECUTE RBS $
$
$
$ compute blade aerodynamics with RBA & BLM module
$
$
EVALUATE R= 40. * .0254 $ convert inches to meters
PARAM VNU = .17894E-04 CA= 340.29 $
EVALUATE RINF = CA * R / VNU $
PARAM IPRINT=3 NORDER=4 $
$
$
CREATE RBA$
UPDATE NEWU=RBA SOURCE=* $
-ADDR NEWM=MACH OLDM=* FORMAT=4H*RS$ MNR=1$
.1 . 3 . 5 . 7 $
-ADDR NEWM=ALPHA OLDM=* FORMAT=4H*RS$ MNR=1$
-6. -3. 0.0 3. 6. $
END*$

$
$
EXECUTE RBA $
EXECUTE BLM $
UPLIST $
$
$
$ save the tables created up to this point on file BCPLIB
$
$
UNLOAD /BCPLIB/ RBS,RBA,BLM $
$
$
ENDCS $

```

## 8.2 Prediction of the Performance and Loads

Templates 3 and 4 are examples of the use of PRP. There are 2 options to run PRP.

- The first option is to input the correct root pitch angle then PRP will compute the power coefficient and other parameters.
- The second option is to input measured power coefficient and an initial guess of blade pitch which is computed as follows:

$$\beta_{75} = \tan^{-1} \frac{V_{\infty}}{\Omega r_{75}}$$

An iterative process is required to obtain the correct root pitch angle to match the measured power. PRP is executed until the computed power coefficient and the measured value match. If measured loads are used for the noise prediction instead of the PAS predicted loads, a table PLD(LOADS) is required. Templates 5 and 6 are examples for computing the loads and using the measured loads table.

### 8.2.1 Template 3 -Execute the Performance Module without Iteration Process

**Problem:** Given the specified flight condition and a library which contains the blade geometry and the sectional lift and drag for set ranges of Mach number and angle-of-attack, compute the induced axial and angular velocities, inflow angle, resultant velocities, power coefficient, thrust coefficient, advance ratio, propeller efficiency, local angle-of-attack, and local Mach number.

**Solution:** The blade element-momentum theory with two-dimensional aerodynamic characteristics of the axially symmetric inflows and induced velocities is used in PRP. In this template, since the blade pitch setting is known, no iterative process is required. The input blade geometry and the lift and drag coefficient tables are stored in a library named BCPLIB which is created from template 2. These tables are used as interpolation tables for a specified flight condition. This specified operating condition has to be in the ranges of angle-of-attack and Mach number computed in template 2.

```

ANOPP $
STARTCS $
$
LOAD /BCPLIB/ RBS RBA BLM $
$
PARAM ALPHAP = 0.      $ set propeller angle-of-attack in degrees
PARAM IDPDT = 0        $ propeller loading is steady
PARAM BETA75 = 19.9     $ propeller 3/4 span pitch angle in degrees
PARAM VF = 51.2         $ flow velocity in m/s
PARAM ORIG = 13.5       $ blade twist angle at 3/4 span in degrees
                        $ (obtain from the blade geometry at 3/4
                        $ span)
EVALUATE BETA = BETA75 - ORIG
                        $ compute root pitch in degrees
EVALUATE THETAR = BETA * PI / 180.
                        $ convert root pitch to radians
EVALUATE ALPHAP = ALPHAP * PI / 180.
                        $ convert propeller angle-of-attack to
                        $ radians
PARAM MACHRF = 0.69     $
PARAM MZ = 0.26         $
$
EXECUTE PRP $
ENDCS $

```

### 8.2.2 Template 4 - Execute the Performance Module with Iteration Process

**Problem:** The power input is known. The blade geometry and the lift and drag coefficient tables are provided from template 1. Compute the performance and find the correct root pitch angle in the specified operating condition.

**Solution:** This template is the same as template 3. The difference is the power input is known in this problem when the root pitch setting is known in template 3. An iterative process is required to find the correct root pitch angle. PRP is executed until the computed power coefficient matches the measured power coefficient.

```

ANOPP JECHO=.TRUE. JLOG=.FALSE. LENG1=20000 $
STARTCS $
$
$
$ this run predicts the noise for the FAA DNW wind tunnel propeller
$ tests. It applies a correction to the propeller blade pitch to
$ match the measured power.
$
$ the following parameters set the tunnel and propeller operating

```

```

$ conditions:
$
$
PARAM ALPHAP = 0.      $ set propeller angle-of-attack in degrees
PARAM IDPDT = 0        $ propeller loading is steady
PARAM BETA75 = 19.9    $ initial guess for propeller 3/4 span pitch
                        $ angle in degrees
PARAM RPM = 2100.      $ propeller rpm
PARAM TEMP = 15.6      $ temperature in degrees Celsius
PARAM POW = 95.9       $ measured power in kilowatts
PARAM VF = 51.2        $ flow velocity in m/s
EVALUATE R = 40./12.   $ blade length in meters
PARAM ORIG = 13.5      $ blade twist from root to 3/4 span
PARAM RHOA = 1.194     $ ambient density in kg/m**3
PARAM IUNITS = 2HSI    $ metric units
PARAM IPRINT = 1       $ request input and output print
PARAM IMPROV = .TRUE.  $ use the improved version of PAS
$
$
$ blade shape is specified by loading library /BLDGEOM/
$
$
LOAD /BLDGEOM/ IBS IBA IBL $
$
$
$ evaluate control statements are used to compute additional required
$ quantities
$
$
EVALUATE RPS = RPM / 60. $ compute revolutions per second
PARAM PI = 3.1415926    $ set value of pi
EVALUATE R = R * 0.3048 $ radius in meter/sec
EVALUATE D = R * 2.     $ compute propeller diameter
EVALUATE CPREF = POW / RHOA / RPS**3 / D**5
                        $ compute power coefficient
EVALUATE BETA = BETA75 - ORIG
                        $ compute root pitch in degrees
EVALUATE THETAR = BETA * PI / 180.
                        $ convert root pitch to radians
EVALUATE ALPHAP = ALPHAP * PI / 180.
                        $ convert propeller angle-of-attack to radians
EVALUATE TA = 1.8 * TEMP + 32.
                        $ convert temperature in degrees Celsius to
                        $ degrees Fahrenheit
EVALUATE CAE = 49. * SQRT ( TA + 459.6 )
                        $ compute speed of sound
EVALUATE CA = CAE * .3048
                        $ speed of sound in meter/sec
EVALUATE MZ = VF / CA
                        $ compute forward mach number
EVALUATE OMEGA = 2. * PI * RPS
                        $ compute angular velocity
EVALUATE MACHRF = R * OMEGA / CA
                        $ compute rotational tip Mach number
$
$
$ the computational grid on the blade surface is now defined
$

```

```

$
UPDATE NEWU=GRID SOURCE=* $
  -ADDR OLDLM=* NEWM=XI1 FORMAT=4H*RS$ $
    0.30    0.35    0.40    0.45    0.50    0.55    0.60    0.65
    0.70    0.75    0.775   0.80    0.825   0.85    0.875   0.90
    0.925   0.95    0.975   0.997    $
  -ADDR OLDLM=* NEWM=XI2 FORMAT=4H*RS$ $
    0.00    0.05    0.10    0.15    0.20    0.25    0.30    0.35    0.375   0.40
    0.425   0.45    0.46    0.47    0.475   0.48    0.485   0.49    0.495   0.50
    0.505   0.51    0.515   0.52    0.525   0.53    0.54    0.55    0.575   0.60
    0.625   0.65    0.70    0.75    0.80    0.85    0.90    1.00    $
  -ADDR OLDLM=* NEWM=PSI FORMAT=4H*RS$ $
    0.00    $
END* $

$
$ the 3/4 span blade pitch must be adjusted so that the computed power
$ coefficient matches the measured power. This requires an iterative
$ solution in the propeller performance (PRP) module. The secant method
$
$ is used to find the root to the equation  $F(Z) = CPREF - CP$ 
$ convergence is assumed when the computed value is within one percent
$ of the measured value.
$
$
  PARAM Z1 = THETAR $
  EXECUTE PRP $
  EVALUATE FZ1 = CPREF - CP $
  EVALUATE THETAR = THETAR + PI / 180. $
  PARAM Z2 = THETAR $
  PARAM COUNT = 1 $
LAB1 CONTINUE $
  EXECUTE PRP $
  EVALUATE FZ2 = CPREF - CP $
  EVALUATE DIFF = FZ2 / CPREF $
  EVALUATE DIFF = ABS(DIFF) $
  EVALUATE COUNT = COUNT + 1 $
  IF ( DIFF .LT. 0.01 ) GOTO LAB2 $
  EVALUATE Z = Z2 - FZ2 * ( Z2 - Z1 ) / ( FZ2 - FZ1 ) $
  PARAM Z1 = Z2 $
  PARAM Z2 = Z $
  PARAM FZ1 = FZ2 $
  PARAM THETAR = Z $
  EVALUATE COUNT = COUNT + 1 $
  IF ( COUNT .GT. 10 ) GOTO LAB3 $
  GOTO LAB1 $
LAB2 CONTINUE $
UNLOAD /PRPLIB/ $
GOTO LAB4 $
LAB3 CONTINUE $
LAB4 CONTINUE $
ENDCS $

```

### 8.2.3 Template 5 - Use PAS to Compute Loads

**Problem:** From the blade shape and the performance libraries, compute the pressure and friction loadings for the noise prediction.

**Solution:** This template is an example of computing the loads using ANOPP PAS. The input for the PLD module are the blade shape and the performance libraries which are created by the blade shape, aerodynamic, boundary layer, and propeller performance modules. PLD is executed to create a load table based on blade element theory together with two-dimensional aerodynamic characteristics.

```
ANOPP JECHO=.TRUE. JLOG=.FALSE. LENG=20000 $
STARTCS $
$
$
$ blade shape is specified by loading library /BLDGEOM/
$
$
LOAD /BLDGEOM/ IBS IBA IBL $
$
$ load the performance results
$
LOAD / PRPLIB/ $
$
$ the following parameters set the tunnel and propeller operating
$ conditions:
$
$
PARAM NBLADE = 2          $ number of propeller blades
PARAM IUNITS = 2HSI       $ metric units
PARAM IPRINT = 3          $ request input and output print
PARAM IMPROV = .TRUE.     $ use the improved version of PAS
$
$
EXECUTE PLD               $ compute loads
$
UNLOAD /PLDLIB/ $
$
ENDCS $
```

### 8.2.4 Template 6 - Experimental Loads for Input

**Problem:** Compute the noise for 2 observers from measured loads for the propeller in template 1.



**Solution:** This is an example of inputting a loads table for the noise prediction instead of computing the loads in PAS. A loads table as a function of spanwise station, chordwise station, and in-plane angle is constructed for the input. Note that there are two order position parameters. The first one is the pressure loading and the second one is the skin friction loading.

```

ANOPP $
STARTCS $
$
$ create table PLD(LOADS)
$
TABLE PLD          (LOADS      ) 1 SOURCE=* $
INT  =              0    1    2
IND1 = RS    9    2    2
      0.5000  0.6000  0.7000  0.8000  0.8500  0.9000  0.9500
      0.9750  0.9970
IND2 = RS   12    2    2
      0.0000  1.2566  1.8850  2.5133  3.0788  3.1102  3.1416
      3.1730  3.7071  4.3982  5.0265  6.2832
IND3 = RS    1    1    1
      0.0000
IND4 = 0      2    0    0
DEP  = RS
      0.0607  0.0728  0.0649  0.0718  0.0907  0.1129  0.1279  0.0114
      0.0126 -0.0492 -0.0531 -0.1025 -0.1491 -0.1600 -0.1681 -0.1873
     -0.4114 -0.4462 -0.1942 -0.2881 -0.2604 -0.2728 -0.3350 -0.4054
     -0.4530 -0.0585 -0.0724  0.0065 -0.0038 -0.1310 -0.2345 -0.2339
     -0.2190 -0.2257 -0.8863 -1.0035  0.1491  0.1935  0.0759 -0.0047
      0.0492  0.1350  0.1945  0.3967  0.4376  0.1501  0.1947  0.1345
      0.0997  0.1426  0.2062  0.2709  0.3801  0.4327  0.1511  0.1960
      0.1932  0.2041  0.2360  0.2774  0.3473  0.3634  0.4279  0.1484
      0.1962  0.1907  0.2084  0.2471  0.2947  0.3547  0.0334  0.1229
      0.0307  0.0412  0.0617  0.0869  0.1035  0.1190  0.1158  0.0855
      0.1141  0.0069  0.0399  0.0542  0.0726  0.0842  0.0959  0.1018
      0.0654  0.0792  0.0206  0.0296  0.0351  0.0441  0.0517  0.0596
      0.0635  0.0385  0.0462  0.0625  0.0775  0.0698  0.0786  0.1000
      0.1251  0.1416 -0.0249 -0.0258  0.0005  0.0007  0.0009  0.0011
      0.0013  0.0014  0.0016  0.0017  0.0023  0.0006  0.0008  0.0010
      0.0012  0.0014  0.0015  0.0017  0.0018  0.0025  0.0007  0.0009
      0.0011  0.0014  0.0016  0.0017  0.0019  0.0021  0.0028  0.0009
      0.0011  0.0014  0.0018  0.0020  0.0023  0.0025  0.0027  0.0037
      0.0005  0.0006  0.0009  0.0012  0.0014  0.0016  0.0017  0.0013
      0.0018  0.0003  0.0003  0.0004  0.0006  0.0007  0.0008  0.0009
      0.0007  0.0009  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
      0.0000  0.0000  0.0000  0.0003  0.0004  0.0005  0.0006  0.0007
      0.0008  0.0008  0.0009  0.0012  0.0009  0.0012  0.0015  0.0019
      0.0021  0.0024  0.0026  0.0029  0.0039  0.0007  0.0009  0.0011
      0.0014  0.0016  0.0017  0.0019  0.0021  0.0028  0.0006  0.0008
      0.0010  0.0012  0.0014  0.0015  0.0017  0.0018  0.0025  0.0005
      0.0007  0.0009  0.0011  0.0013  0.0014  0.0016  0.0017  0.0023
END* $
$

```

```

$
$  load the blade shape library
$
$
LOAD /BLDGEOM/  $
$
$
PARAM IMPROV = .TRUE  $ use the improved version of PAS
PARAM      R = 1.016  $ blade length in meters
PARAM NBLADE = 2      $ number of propeller blades
PARAM  RHOA = 1.194   $ ambient density in kg/m**3
PARAM   CA  = 340.0   $ speed of sound in m/s
PARAM   RPM = 2100    $ propeller rpm
PARAM   VF  = 51.2    $ flow velocity in m/s
PARAM THETAR = 0.1292 $ root pitch angle in rad.
PARAM  NHARM = 20     $ number of harmonics desired
PARAM  NTIME = 512    $ number of time points for waveform
PARAM IUNITS = 2HSI   $ metric units
PARAM  IATM = 0       $ atmospheric data from user parameters
PARAM  IOUT = 0       $ no output unit member
PARAM   PI  = 3.1416  $
EVALUATE RPS = RPM / 60.  $ revolutions per second
EVALUATE MZ = VF / CA    $ compute forward Mach number
EVALUATE OMEGA = 2. * PI * RPS
                        $ compute angular velocity
EVALUATE MACHRF = R * OMEGA / CA
                        $ compute rotational tip Mach number
$
$ observer positions are defined for the two observers of interest
$
UPDATE NEWU=OBSERV SOURCE=* $
      -ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
      4.453  0.  2.571  $ observer 1    30 degrees
      4.000  0.  0.     $ observer 2    0 degrees
END* $
$
EXECUTE SPN  $
$
ENDDCS  $

```

### 8.3 Template 7 - Near-Field Noise Prediction

**Problem:** Predict the noise for the two observers having x,y,z coordinates with respect to the propeller hub given by

4.453 m	0. m	2.571m	observer 1
4.000 m	0. m	0. m	observer 2

at the following operating condition:

propeller RPM	=	2100
number of blades	=	2
blade length	=	1.016 m
inflow velocity	=	51.2 m/s
root pitch angle	=	0.1292 rad.

**Solution:** The blade geometry and the blade surface pressure are known from template 1 and template 5. The computation of the periodic acoustic pressure signature and the spectrum of the propeller with subsonic tip speed are based on a solution of the Ffowcs Williams-Hawkings equation without the quadrupole source term. The observers are assumed to be moving with the aircraft and the full blade approximation is used in this prediction.

```
ANOPP $
STARTCS $
$
$ observer positions are defined for the two microphones of interest
$
UPDATE NEWU=OBSERV SOURCE=* $
-ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
4.453 0. 2.571 $ observer 1 30 degrees from propeller plane
4.000 0. 0. $ observer 2 0 degrees or in the propeller plane
END* $
$
$ blade shape is specified by loading library /BLDGEOM/
$
LOAD /BLDGEOM/ IBS IBA IBL $
$
$ Pressure and skin loadings in the /PLDLIB/ library
$
UNLOAD /PLDLIB/ $
$
PARAM IMPROV = .TRUE $
PARAM R = 1.016 $ blade length in meters
PARAM NBLADE = 2 $ number of propeller blades
PARAM RHOA = 1.194 $ ambient density in kg/m**3
PARAM CA = 340.0 $ speed of sound in m/s
PARAM RPM = 2100 $ propeller rpm
```

```

PARAM      VF = 51.2      $ flow velocity in m/s
PARAM THETAR = 0.1292    $ root pitch angle in rad.
PARAM  NHARM = 20        $ number of harmonics desired
PARAM  NTIME = 512       $ number of time points for waveform
PARAM IUNITS = 2HSI      $ metric units
PARAM  IATM = 0          $ atmospheric data from user parameters
PARAM  IOUT = 0          $ the default value for iout is 0. For near
                        $ -field noise prediction, iout=0 or iout=2
                        $ and OBSERV(COORD) are required. If
                        $ SPN(FFT) and SPN(TIME) are required, then
                        $ iout=2
PARAM      PI = 3.1416    $
EVALUATE RPS = RPM / 60.  $ revolutions per second
EVALUATE MZ = VF / CA     $ compute forward mach number
EVALUATE OMEGA = 2. * PI * RPS $ compute angular velocity
EVALUATE MACHRF = R * OMEGA / CA
                        $ compute rotational tip Mach number
PARAM METHOD = 1          $ full blade surface approximation
EXECUTE SPN $
ENDCS $

```

## 8.4 Template 8 - Noise Bubble for Far-Field Noise Prediction

**Problem:** This problem is similar to the problem in template 7 with the exception that observer coordinates are input in the spherical format. In this example, the observers are on the plane perpendicular to the propeller plane starting from the flight direction to the back of the propeller at the following angles:

$$\begin{array}{cccccccc} 10^\circ & 30^\circ & 50^\circ & 90^\circ & 120^\circ & 150^\circ & 179^\circ \\ \text{Source radius} & = & & 5 * R & & \text{where } R \text{ is the propeller radius} \end{array}$$

**Solution:** The same computational method is used in template 7. In this template, observers are defined on an imaginary bubble which is specified by a constant radius, polar directivity angles, and the azimuthal angles. The polar directivity angle is specified from the front of the propeller (0°) to the back of the propeller (180°) in the inflow direction. The azimuthal angle is determined from the left hand side to right hand side of the propeller specified from the inflow direction looking toward the propeller. A noise table is created on the bubble which is used later by way of interpolation to evaluate the noise at specified observers on the ground. It is important to set the parameter IOUT=2 and to input the unit member SFIELD(THETA) and SFIELD(PHI). The source radius RX, re R is also required.

```

ANOPP $
STARTCS $
$
$ the sound field arrays must be defined. In this case observers are
$ chosen only on the propeller plane
$
UPDATE NEWU=SFIELD SOURCE=* $
  -ADDR OLDLM=* NEWM=THETA FORMAT=4H*RS$ $
    10. 30. 50. 90. 120. 150. 179. $
  -ADDR OLDLM=* NEWM=PHI FORMAT=4H*RS$ $
    0. .1 .2 .3 .4 .5 .6 .7 .8 .9 1. $
$
END* $
$
$ additional output control parameters are required
$
PARAM IATM = 0 $ use atmospheric user parameters
PARAM IOUT = 1 $ generate farfield noise table in SPN
PARAM IDPDT = 1 $ unsteady because of angle-of-attack
PARAM ALPHAP = 0.020 $ propeller angle-of-attack in radians
PARAM R = 1.016 $ blade length in meters
PARAM MZ = 0.157 $ flight Mach number
PARAM THETAR = 0.0576 $ root pitch angle in radians
PARAM MACHRF = 0.649 $ rotational tip Mach number
PARAM CA = 342.5 $ speed of sound in meters/sec
PARAM RHOA = 1.162 $ density in kg/meters**3

```

```
$  
$ the blade shape library is BLDGEOM and the load library is PLDLIB  
$  
LOAD / BLDGEOM/ $  
LOAD /PLDLIB/ $  
$  
EXECUTE SPN $  
UNLOAD /SPNLIB/ $  
ENDCS $
```

## 8.5 Flyover Noise Prediction

### 8.5.1 Template 9 - One Propeller

**Problem:** Given observers on the ground, predict the noise for the observers when the aircraft is at a level flyover with the following operating conditions:

Aircraft speed	=	51.2 m/s
Flight path angle	=	6.2°
Altitude	=	211.5 m
Propeller angle-of-attack	=	1.15 °

The propeller operating condition is the same as template 8.

**Solution:** Atmospheric properties for the given altitude will be determined from using the standard atmospheric table. This table is in PROCLIB library as shown below.

The first step is to create a noise bubble as it was done in template 8. The second step is to find the position of the noise source as function of time in the Steady Fly Over (SFO) Module. The third step is to execute GEO to establish the vectors from the source to the observer. The output from GEO includes the polar directivity angle, the azimuthal angle, and the elevation angle as functions of reception time. The fourth step is to execute PRT. PRT will sum the noise at the same frequencies if there are more than one noise source (correlated). Then the noise is computed at the observer location with information provided by the GEO module. Doppler shift, spherical spreading, and characteristic impedance effects are always included in the calculations. Options are available for atmospheric absorption and ground effects. Module LEV computes noise metrics (A-weighted, etc.). Lastly, the Effective Noise Module (EFF) computes EPNL if requested.

```
ANOPP $
STARTCS $
$
$ standard atmosphere is loaded from system library load the noise
$ library
$
$
LOAD /LIBRARY/ PROCLIB $
CALL PROCLIB (ATMSTD) $
LOAD /SPNLIB/ $
$
$ the flight path is now defined using the steady flyover module
$
PARAM VF = 51.20 $ aircraft speed in m/s
PARAM PATHANG = 6.2 $ flight path angle in degrees
PARAM VI = VF $ set forward speed
PARAM ENGNAM = 3HXXX $ set member name parameter
```

```

PARAM      TI = -20.          $ set start time in seconds
PARAM      TSTEP = 1.0        $ set time step in seconds
PARAM      THW = PATHANG      $ set path angle
EVALUATE    XI = TI * VI      $ compute starting x position
PARAM      YI = 0.            $ set starting y position
EVALUATE    ZI = 211.5 + XI * SIN(THW)
                                $ compute starting z position
PARAM      TF = 10.           $ set final time in seconds
EVALUATE    XF = TF * VI      $ compute final x position
PARAM      ZF = ZI            $ final z position
PARAM      ALPHA = 1.15       $ propeller angle-of-attack in degrees
$
$ execute SFO module using default values for remaining parameters
$
EXECUTE SFO $
$
$ reset parameters for geometry module
$
PARAM      START = -20.        $ start time in seconds
PARAM      STOP = TF           $ ending time in seconds
PARAM      DELT = 0.5          $ reception time increment in seconds
PARAM      DELDB = 20.         $ limiting noise level, down from the
                                $ peak (dB)
PARAM      ICOORD = 1          $ request body axis output only
$
$
$ The remaining parameters use default values. The location of the
$ observers are input in meters referenced to the point 2500 meters
$ from brake release. The first microphone (observer) position is at
$ that point. The second observer position is at 1890 meters from
$ brake release which corresponds to an X coordinate of -610 meters.
$ Both flush and 1.2 meter microphones were used.
$
$ The observer input member is:
$
UPDATE NEWU=OBSERV SOURCE=* $
      -ADDR OLDLM=* NEWLM=COORD FORMAT=4H3RS$ $
          0.      0.      0. $ primary mic - ground
          0.      0.      1.2 $ primary mic - 1.2 meter
        -610.     0.      0. $ secondary mic - ground
        -610.     0.      1.2 $ secondary mic - 1.2 meter
END* $
$
$ execute geometry module
$
EXECUTE GEO $
$
$ now, the parameters are defined for the tone propagation (PRT)
$ module
$
PARAM      R = 1.016           $ blade in meters
PARAM      RX = 5.             $ pick source radius to be 5
                                $ propeller radii
PARAM      NBLADE = 2          $ number of blades
PARAM      RPM = 2100.         $ propeller rpm
EVALUATE    RPS = RPM / 60.    $ compute revolutions/second
EVALUATE    DELF = RPS * NBLADE $ bandwidth is blade passing

```



```

EVALUATE    RS = RX * R           $ frequency
                                           $ convert to dimensional source
                                           $ radius
PARAM SURFACE = 4HSOFT           $ soft boundary
PARAM ABSORP = .TRUE.           $ predict standard absorption effect
PARAM GROUND = .TRUE.           $ predict ground effects
$
$ tone propagation module is executed with remaining parameters
$ defaulted
$
EXECUTE PRT YYYYYY=SPN GEOM=BODY $
$
$ the noise levels module (LEV) is executed to compute frequency
$ integrated levels. Only narrow band levels are computed.
$
PARAM      NAWT=.TRUE.      NDWT=.TRUE.      NOSPL=.TRUE.
                                           $ set narrow band level flags to true
PARAM      IAWT=.FALSE.     IDWT=.FALSE.     IOSPL=.FALSE.     IPNL=.FALSE.
      IPNLT=.FALSE.        $ set 1/3-octave band level flags to false
PARAM MEMSUMN=4HPRT  4HPRES
                                           $ define input member
$
$ the lev module is now executed
$
EXECUTE LEV $
$
EXECUTE EFF $
$
ENDCS $

```

### 8.5.2 Template 10 - Tilt Rotor

**Problem:** Predict the noise for the tilt rotor in propeller mode. This tilt rotor has two identical propellers, the NACA 16 airfoil. The aerodynamic information such as lift and drag coefficients are built in the library NACALB. The operating conditions are

Flight altitude	=	250 ft
Indicated airspeed	=	150 knots
Number of blades	=	4
Propeller radius	=	4 ft
Propeller RPM	=	1550.

Distance from each propeller hub to the centerline of the aircraft is 12.5 ft

**Solution:** This example is a flyover noise prediction for a tilt rotor in the propeller mode. SPN creates a noise bubble for each propeller. A common point between the two

propellers is chosen. The MSN module sums the noise of the two propellers at a common point. The trailing edge noise for each propeller is also computed from PTE. After SFO establishes the source location, GEO establishes the source to observer geometry, PRT propagates the tone noise from MSN and PRO propagates the broad band noise from PTE to chosen observers on the ground. Then LEV sums the total noise.

```

ANOPP  $
STARTCS $
$
$-----$
$
$                                USER INPUT PHASE OF RUN
$
$-----$
$
PARAM  ALT      = 250.      $ altitude in feet
PARAM  VIAS     = 150.      $ indicated airspeed in knots ( this
                           $ should be changed to true airspeed when
                           $ it is available )
PARAM  VGS      = 150.0     $ ground speed in knots
PARAM  TILT     = 0.        $ nacelle tilt angle in degrees
PARAM  PSI      = 180       $ direction of flight ( affects microphone
                           $ numbering )
PARAM  IUNITS   = 7HENGLISH $ English units are being used
PARAM  NBLADE   = 4         $ four propeller blades
PARAM  RADIUS   = 4.00      $ propeller radius in feet
PARAM  IPRINT   = 1         $ turn off output print to save paper
$
$ NACALB is the blade shape library which was created from the
$ execution of IBS, IBA, and IBL from using NACA 16 airfoil.
$ The standard atmospheric table in the library named LIBRARY
$
$-----$
LOAD /NACALB/ $
LOAD /LIBRARY/ ATM=ATMOS (AAC=ABS TMOD=STRD) $
$-----$
$
$ the propeller positions are defined for the multirotor source noise
$ module so the acoustic interaction effects between the propellers can
$ be determined. Note that change from propeller to rotor coordinates
$ does not affect this input
$
PARAM  X1       = 0. 12.500 0. $ position of first propeller
PARAM  X2       = 0. -12.500 0. $ position of second propeller
$
$-----$
$                                COMPUTATIONAL GRIDS
$-----$
$
$ the directivity angle and observer position data are entered here.
$ The standard ANOPP grids are used.
$
$
UPDATE NEWU=SFIELD SOURCE=* $
      -ADDR OLDLM=* NEWLM=THETA FORMAT=4H*RS$ $

```

```

    10.  30.  60.  90. 120. 150. 179. $
-ADDR OLDLM=* NEWM=PHI FORMAT=4H*RS$ $
    -90. -75. -60. -45. -30. -15.  0.
    15.  30.  45.  60.  75.  90.  $
-ADDR OLDLM=* NEWM=FREQ FORMAT=4H*RS$ $
    12.5 16.  20.  25.  31.5 40.  50.  63.  80.
    100. 125. 160. 200. 250. 315. 400. 500. 630.
    800. 1000. 1250. 1600. 2000. 2500. 3150. 4000. 5000.
    6300. 8000. 10000.  $
END* $
$
$ define microphone positions based on flight direction
$
IF ( PSI .EQ. 0 ) GOTO LAB10 $
$
$ microphone definitions for 180 degree direction of flight (start
$ from the opposite direction of flight toward the direction of flight)
$
UPDATE NEWU=OBSERV SOURCE=* $
-ADDR OLDLM=* NEWM=COORD FORMAT=4H3RS$ $
    0. -209.77  0.  $ ROW2
    0. -144.34  0.  $
    0. -90.99  0.  $
    0.  0.  0.  $ ROW3
    0.  90.99  0.  $
    0. 144.34  0.  $
    0. 209.77  0.  $
END* $
$
GOTO LAB20 $
LAB10 CONTINUE $
$
$ microphone positions for 0 degree direction of flight (start from
$ the direction of flight to the opposite direction of flight)
$
UPDATE NEWU=OBSERV SOURCE=* $
-ADDR OLDLM=* NEWM=COORD FORMAT=4H3RS$ $
    0. 209.77  0.  $ ROW2
    0. 144.34  0.  $
    0. 90.99  0.  $
    0. 0.  0.  $ ROW3
    0. -90.99  0.  $
    0. -144.34  0.  $
    0. -209.77  0.  $
END* $
$
LAB20 CONTINUE $
$
$ The computational grid on the propeller disk for performance and
$ noise calculations is entered next. The following grid works well
$ for all cases to date.
$
$
UPDATE NEWU=GRID SOURCE=* $
-ADDR OLDLM=* NEWM=XI1 FORMAT=4H*RS$ $
0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.95 0.98 1.00 $
-ADDR OLDLM=* NEWM=XI2 FORMAT=4H*RS$ $
    0. 0.05 0.15 0.25 0.35 0.40

```

```

0.45 0.50 0.55 0.60 0.625 0.65
0.70 0.75 0.80 0.85 0.90 0.95 1.00 $
-ADDR OLDLM=* NEWM=PSI FORMAT=4H*RS$ $
0.00 $
END* $
$
$-----$
$
$ DERIVED AND STANDARD USER PARAMETERS
$-----$
$
$
PARAM IMPROV = .TRUE. $ improved blade section modules are used
PARAM CA = 1115.49 $ speed of sound in ft/sec
PARAM ALPHAP = TILT $ set propeller angle-of-attack to tilt
$ angle
PARAM IDPDT = 0 $ propeller loading is steady
PARAM NBLADE = 4 $ number of propeller blades
PARAM PI = 3.1415927 $ define value of pi
PARAM RPM = 1550. $ propeller RPM
EVALUATE OMEGA = RPM * PI / 30. $ compute propeller angular velocity
EVALUATE MACHRF = OMEGA * RADIUS / CA $ compute propeller hover tip Mach number
EVALUATE KNTFPS = 0.5144444 / .3048 $ conversion factor from knots to
$ feet/sec
EVALUATE MZ = VIAS * KNTFPS / CA $ compute propeller relative forward
$ speed
EVALUATE D = RADIUS * 2. $ compute propeller diameter
EVALUATE RPS = RPM / 60. $ compute revolutions per second
$
$ here the inputs for the first execution of the MSN module are set up
$
EVALUATE RS = 10. * RADIUS $ compute dimensional observer distance
PARAM R1 = RADIUS $ radius of first propeller
PARAM R2 = RADIUS $ radius of second propeller
PARAM IOPT = 2 $ propeller coordinate system option
$
$ SPN module
$
PARAM METHOD = 1 $ use line source method to save time
PARAM IOUT = 2 $ request near field noise analysis
PARAM NTIME = 128 $ use 128 time points in time history
PARAM NHARM = 16 $ request 32 harmonics
PARAM RX = 5. $ set propeller source radius to five
$ diameters
$
$ SFO module
$
PARAM ENGNAM = 3HXXX $ change value of engnam to match default
$ values of other modules
PARAM TSTEP = 1. $ data generated at 1 second intervals
EVALUATE VI = VGS * KNTFPS $ convert forward speed to feet/sec
EVALUATE XI = - ALT / SIN( 15. ) $ start flight path at polar directivity
$ of 15 degrees

```

```

EVALUATE START = XI / VI      $ set start time of flight path
PARAM    TI    = START        $ start is a geo parameter - it is
                                $ changed by SFO so both are set here
EVALUATE XF    = -1. * XI     $ final flight path distance set at same
                                $ distance past overhead
EVALUATE TF    = -1. * START  $ corresponding final time
PARAM    VF    = VI          $ final velocity
$
$  GEO module
$
EVALUATE AW    = PI * RADIUS**2
                                $ aircraft reference area in square feet
PARAM    WEIGHT = 12872.      $
EVALUATE MASSAC = WEIGHT / 32.174
                                $ compute mass of aircraft in slugs
PARAM    DTIME  = 0.5        $ compute in 1/2 second increments for
                                $ EPNL calculations
PARAM    ICOORD = 1          $ request body axis coordinate system
$
$  PRO and PRT modules
$
EVALUATE DELF  = OMEGA * NBLADE / 2. / PI
                                $ the propeller bandwidth is the
                                $ fundamental frequency
PARAM    SURFACE = 4HHARD    $ microphones were on hard surfaces
PARAM    ABSORP  = .TRUE.    $ request atmospheric absorption effects
                                $ be applied to the data
PARAM    GROUND  = .TRUE.    $ request ground effects be applied to the
                                $ data
$
$  LEV module
$
PARAM    IAWT    = .TRUE.    $
PARAM    IDWT    = .FALSE.   $
PARAM    IOSPL   = .TRUE.    $
PARAM    IPNL    = .TRUE.    $ set flags for metric calculations
PARAM    IPNLT   = .TRUE.    $
PARAM    NAWT    = .FALSE.   $
PARAM    NDWT    = .FALSE.   $
PARAM    NOSPL   = .FALSE.   $
$
$  bring the load libraries
$
LOAD / PLDLIB/ $
$
$-----
$              MODULE EXECUTION PHASE OF RUN
$-----
$
$  now, the MSN module is executed
$
EXECUTE MSN OBSERV=DUMMY $
$
$  the SPN module is executed for the first propeller using the name
$  overrides for the first propeller.
$
$
$
$

```

```

EXECUTE SPN R=RADIUS TIME=HIST1 OBSERV=MSN COORD=OBS1 $
$
$ the SPN module is executed for the second propeller using the name
$ overrides.
$
PARAM    PSIO      = 0.0      $ set blade offset to 0 radian
PARAM    ROTLEFT   = .TRUE.   $ change direction of rotation
$
EXECUTE SPN R=RADIUS TIME=HIST2 OBSERV=MSN COORD=OBS2 $
$
$ now the MSN module is executed again to sum the two sources
$
PARAM    IMODE      = 2        $ set mode to sum sources
$
EXECUTE MSN TIM=SPN OBSERV=DUMMY $
$
$ the PTE module is now executed for the right propeller. The
$ direction of rotation must be switched back to right handed.
$
PARAM    ROTLEFT    = .FALSE.  $ request right hand rotation
PARAM    IOUT        = 1       $ reset IOUT for PTE execution
$
EXECUTE PTE R=RADIUS $
$
$ the PTE module is now executed for the left propeller. The
$ direction of rotation must be switched back to left handed.
$
PARAM    ROTLEFT    = .TRUE.   $ request left hand rotation
$
EXECUTE PTE R=RADIUS PTE=PTE2 $
$
$ the main propeller angle-of-attack must be converted to degrees
$ before execution of the SFO module
$
$ SFO is now executed
$
EXECUTE SFO ALPHA=ALPHAP ZI=ALT $
$
$ the geo module is now executed
$
EXECUTE GEO STOP=TF $
$
$ the broadband noise is now propagated by the PRO module.
$
PARAM    PROSUM      = 4HPTE   4HPTE2
$ propagate propeller broadband noise - it is summed to account for
$ both propellers
$
$ execute PRO for propeller
$
EXECUTE PRO GEOM=BODY $
$
$ the tone propagation module is now executed to propagate the tone
$ noise source to the observer.
$
EXECUTE PRT YYYYYY=MSN GEOM=BODY $
$
$ the remaining statements will be executed in this job and also in

```

```

$ the restart job that propagates tone noise predictions. The noise
$ levels (LEV) module is now executed to compute the frequency
$ integrated levels and to sum noise sources. The parameter MEMSUMN
$ has the unit member name of the PRT output to be summed. Note
$ that all names in MEMSUMN must have the same hollerith length.
$ Similarly the user parameter memsum has the names of the PRO output.
$
PARAM      IPRINT      = 3          $ turn on output print for total noise
PARAM      MEMSUMN      = 4HPRT  4HPRES
                                $ propagate tone noise source
PARAM      MEMSUM       = 4HPRO  4HPRES
                                $ propagate broadband noise source
$
$ the lev module is now executed
$
EXECUTE LEV LEV=LEVTOT $
$
$ the source summed 1/3 octave band spectra are written to the
$ external file finally the effective noise module is executed
$
EXECUTE EFF LEV=LEVTOT $
$
$ the job is finished
$
PROCEED $
ENDCS $

```

## 9. PAS Prediction and Measured Data

### 9.1 PAS Prediction Results

#### 9.1.1 Four Methods from SPN

The results from using the different options in the Subsonic Propeller Noise (SPN) module is presented and compared with measured data. In general, the predictions from the four methods are very close for the first few harmonics, but for higher harmonics, the discrepancy increases. These options permit greater compatibility with users computer resources. The full blade surface method uses the most CPU time compared to the other methods and the point source approximation uses the least amount of CPU time.

Figure 7 illustrates the DNW tunnel configuration along with 2 chosen microphones. Predictions from the four methods are shown in figures 8, 9 and 10. Data available for the comparison is obtained from reference 8. Operating conditions for the predictions are as follows:

Run number	RPM	Flow Vel. m/s	Attitude Angle (deg.)	Total Mach number
BN4	2100	51.2	0.	0.67
GN3	2700	77.0	-7.4	0.87
EN2	2400	51.9	7.3	0.77

#### 9.1.2 Synchrophasing Using PAS

Synchrophasing was studied using PAS to determine the effects of blade phase angle for a tilt rotor in the propeller mode. In template 9, note that there is one blade offset parameter name PSI0. This parameter is to input the blade offset angle (phase) for the study of the effect of the blade phasing angle in the tilt rotor case.

##### Propeller geometry:

- NACA 16 series blade
- 8 feet diameter
- 25 ft hub to hub separation
- Each propeller has 4 blades



Flight conditions:

- Altitude = 250 feet
- Velocity = 150 Knots
- RPM = 1500
- $C_p$  = 0.024

Effect of blade phasing on:

- Sound exposure level, SEL

The purpose of this study is to determine the effect of varying the phase angle from  $0^\circ$  to  $20^\circ$  on twin engine propeller noise. Maximum dBA and sound exposure level (SEL) are computed in the study.

The following table and figures are enclosed for the results of this study:

- \* Figure 11 shows the source observer geometry.
- \* Figure 12 shows the relative rotation of the propellers. For case 1, the starboard propeller rotates clockwise as viewed from the back to the front of the aircraft. The port propeller is phased from  $0^\circ$  to  $20^\circ$  in  $5^\circ$  increments. Case 2 is similar except that the starboard propeller rotates counter clockwise and the port propeller rotates clockwise.
- \* Figure 13 shows the effect on SEL of the blade phasing for each observer.

## 9.2 Comparison of PAS Prediction and DNW Data

Predictions from SPN were compared with DNW data for both the round-tip and square-tip propellers. The chosen DNW runs are as follows for the square-tip propeller

Run No.	Rotation speed RPM	Attitude angle ( $\alpha$ ) Degrees	Flow velocity m/s	Helical Mach No.
CC-3	1800	0	51.2	.5825
BC-4	2100	0	51.2	.6762
BC-5	2400	0	51.5	.7671
BC-6	2700	0	77.0	.8775
LC-1	2100	3.8	51.6	.6760
LC-2	2400	3.8	51.5	.7675
LC-3	2700	3.8	76.9	.8745
LC-4	1800	3.8	51.2	.5840
EC-1	2100	-7.3	51.7	.6752
EC-2	2400	-7.3	51.9	.7667
EC-3	2700	-7.3	76.9	.8733
EC-4	1800	-7.3	51.2	.5831

and for the round-tip propeller:

Run No.	Rotation speed RPM	Attitude angle ( $\alpha$ ) Degrees	Flow velocity m/s	Helical Mach No.
CN-3	1800	0	51.5	.5838
BN-4	2100	0	51.2	.6729
BN-5	2400	0	51.5	.7639
BN-6	2700	0	77.2	.8758
FN-1	2100	-3.6	51.6	.6746
FN-2	2400	-3.6	51.7	.7655
FN-3	2700	-3.6	77.2	.8740
FN-4	1800	-3.6	51.5	.5829

GN-1	2100	7.4	51.4	.6751
GN-2	2400	7.4	51.7	.7664
GN-3	2700	7.4	77.0	.8735
GN-4	1800	7.4	51.2	.5830

The microphone configuration is shown in figure 14. The five chosen microphones are microphones 2, 4, 6, 8, 9. The angles-of-attack for the round-tip propeller are  $0^\circ$ ,  $-3.6^\circ$ , and  $7.4^\circ$ . For the square-tip propeller, the chosen angles-of-attack are  $0^\circ$ ,  $3.8^\circ$ , and  $-7.3^\circ$ . The positive angle means that the propeller is nose down to the microphone array and the negative angle it is nose up.

Figures 15 to 18 show the comparison of the PAS and DNW data for the round-tip propeller for microphones 2 and 4 for 1800, 2100, 2400, and 2700 RPM and the attitude angles are  $0^\circ$ ,  $-3.6^\circ$ , and  $7.4^\circ$ . Figures 19 to 22 are also for the round-tip propeller at the angles-of-attack  $-3.6^\circ$  and  $7.4^\circ$  for microphones 6, 8, and 9. Figure 19 is for 1800 RPM, figure 20 is for 2100 RPM, figure 21 is for 2400 RPM, and figure 22 is for 2700 RPM. Figures 23 to 30 depict the results for the square-tip propeller for the same operating conditions. The comparisons of PAS predictions and DNW data for the round-tip propeller seem to be better than for the square-tip propeller, especially at high RPM.

## REFERENCES

1. Wilson, Mark R., "An Introduction to High Speed Aircraft Noise Prediction," NASA CR - 189582, 1992.
2. Gillian, Ronnie E., Brown, Christine G., Bartlett, Robert W., and Baucom, Patricia H., "ANOPP Programmer's Reference Manual for the Executive System," NASA TMX-74029, 1977.
3. Zorumski, William E., and Weir, Donald S., "Aircraft Noise Prediction Program Theoretical Manual Propeller Aerodynamics and Noise," Part 3, NASA TM - 83199, 1986
4. Nguyen, L. Cathy, "The NASA Aircraft Noise Prediction Program Improved Propeller Analysis System," CR - 4394, 1991.
5. Gillian, Ronnie E., "Aircraft Noise Prediction Program User's Manual," NASA TM-84486, 1983.
6. Zorumski, William E., "Aircraft Noise Prediction Program Theoretical Manual," Parts 1 & 2. NASA TM-83199, 1981.
7. Nolan, Sandra K., "Aircraft Noise Prediction Program Propeller Analysis System IBM-PC Version User's Manual," Version 2.0, CR 181689, 1988.
8. Dobrzynski, Werner M., Heller, Hanno H., Powers, John O., and Densmore, James E., "DFVLR/FAA Propeller Noise Test in the German-Dutch Wind Tunnel DNW," Executive Data Report No. AEE 86-3, 1986.

## Appendix A. Summary of Functional Modules

Module Name	Module Title	Brief Description
ABS	Atmospheric Absorption Module	Computes absorption coefficient as a function of altitude & frequency using ANSI or SAE method
ATM	Atmospheric Module	Computes atmospheric properties as a function of altitude using hydrostatic method
BLM	Boundary Layer Module	Computes skin friction, drag coefficients, and boundary layer thickness at trailing edge using the integral formulations
EFF	Effective Noise Module	Computes the Effective Perceived Noise Levels
GEO	Geometry Module	Calculates source-to-observer geometry
IBA	Improved Blade Aerodynamics Module	Same as RBA with addition of Glauert compressibility correction and increased number of Fourier series terms
IBL	Improved Boundary Layer Module	Same as BLM with additional zero pressure gradient flat plate model for the computation of the boundary layer thickness
IBS	Improved Blade Shape Module	Same as RBS with more concise input blade geometry and produces additional output tables
LEV	Noise Levels Module	Computes OASPL, A-weighted SPL, D-weighted SPL, PNL, and/or PNLT
PLD	Propeller Loading Module	Calculates loads at specified surface points and times
PRO	Propagation Module	Transfers broad-band noise data from the source frame of reference to the observer frame of reference
PRP	Propeller Performance Module	Computes induced velocity field, thrust, torque, and efficiency under specified operating conditions
PRT	Tone Propagation Module	Transfers tone noise data from the source frame of reference to the observer frame of reference
PTE	Propeller Trailing Edge Noise Module	Predicts broad-band and harmonic noise due to the interaction of the blade turbulent boundary layer with the trailing edge
RBA	Blade Aerodynamics Module	Computes pressure forces on the upper and lower surfaces for specified angle-of-attack and Mach numbers
RBS	Blade Shape Module	Formulates a functional representation of the blade surface suitable for aerodynamic and aeroacoustic calculations
SFO	Steady Flyover Module	Provides flight dynamics data for a steady flyover
SPN	Subsonic Propeller Noise Module	Calculates periodic acoustic pressure signature and spectrum with subsonic tip speed
TPN	Transonic Propeller Noise Module	Calculates periodic acoustic pressure signature and spectrum with transonic tip speed

## Appendix B. - TABLE Control Statement Discussion

The TABLE control statement builds an ANOPP data table which can be used as input to the following functional modules. What follows is a brief description of the elements of a table control statement and how these elements fit together to form a usable ANOPP table. For more detailed information refer to Section 3.7.3 of reference 2.

**Format:**           Type 1 Tables (only type currently available).

A table is generally has the following format:

```
TABLE UNIT(MEMBER) 1 SOURCE=* $
INT=0,1,2
IND1=RS,n1,2,2, independent variable values separated by commas or blanks
IND2=RS,n2,2,2, independent variable values separated by commas or blanks
IND3=RS,n3,2,2, independent variable values separated by commas or blanks
IND4=RS,n4,2,2, independent variable values separated by commas or blanks
DEP=RS, dependent variable values separated by commas or blanks
END* $
```

The first word of the first line is TABLE following the data unit member names. Number 1 shows that type one table is currently available. SOURCE= specifies where the data is located from which the table will be built. The \* indicates that the data will immediately follow the TABLE control statement. As with any ANOPP control statement, this line must end with a dollar sign symbol (\$).

The line beginning with INT determines which interpolation procedures will be permitted in this table. A 0 indicates no interpolation, a 1 indicates linear interpolation, and a 2 indicates cubic-spline interpolation.

The next four lines define the independent variables (IND ). The maximum for the independent variable types is four. Each of the independent variable cards has the following descriptions:

- RS : real single precision
- The first number is the number of independent variables in that line
- The second integer number is the interpolation code for the extrapolation procedure to be used if the specified value for this independent variable falls **beyond the last table** value for the independent variable.

- The third integer number is the interpolation code for the extrapolation procedure to be used if the specified value for this independent variable falls **before the first table** value for the independent variable.

These interpolation codes are

- 0 no extrapolation allowed
- 1 use table value of the independent variable closest to the specified value
- 2 extrapolation is linear when using the first two table values.

The next numbers are the independent variables in ascending or descending order.

Multiple dependent variables can be assigned in one ANOPP table structure. To implement a multiple dependent variable table, the ordered position format code is used on an additional independent variable card IND. The additional dependent variables are added to the dependent variable list. In the following example, a drag coefficient table will be added to the lift coefficient table described above.

```

TABLE      BLM(LIFTDRAG)      1      SOURCE=* $
INT      = 1
IND1 = RS      1      1      1      0.
IND2 = RS      9      1      1      -16.0 -12.0 -8.0 -4.0 0.
                                4.0  8.0 12.0 16.0
IND3 = RS      4      1      1      0.      0.25 0.55 0.85
IND4 =      0      2      0      0
DEP      = RS
-1.6      -1.2      -0.8      -0.4      0.      0.4      0.8      1.2      1.6
-1.6      -1.24      -0.83      -0.41      0.      0.41      0.83      1.24      1.6
-1.6      -1.44      -0.96      -0.48      0.      0.48      0.96      1.44      1.6
-1.6      -1.6      -1.52      -0.76      0.      0.76      1.52      1.6      1.6
0.017      0.012      0.009      0.007      0.006      0.007      0.009      0.012      0.017
0.017      0.012      0.009      0.007      0.006      0.007      0.009      0.012      0.017
0.017      0.012      0.009      0.007      0.006      0.007      0.009      0.012      0.017
0.017      0.012      0.009      0.007      0.006      0.007      0.009      0.012      0.017
END*      $

```

In this example, BLM is the data unit and LIFTDRAG is the table member. The table member name must be enclosed in parentheses. The number 1 following the data unit/table member definition indicates that this will be a type-1 data table. Type-1 data tables are the only type of data tables supported by ANOPP at this time. The next line beginning with INT determines which interpolation procedure will be used in the table. In this example, linear interpolation will be permitted on this data table. The lift and drag coefficients of a particular propeller are a function of spanwise station (IND1), angle-of-attack (IND2), and Mach number (IND3). IND4 shows that there are 2 ordered positions. This table consists of two dependent variables as a function of three independent variables.

The IND1 line defines the spanwise station. The character following the IND1= indicates the data-type code for the spanwise station. A value of RS means the values will be real single precision. The next value in this line determines the number of independent variables in this line. There is one value of the spanwise station.

The IND2 is the same as IND1. There are 9 values of the angle-of-attack. The number which follows is an integer code which defines the extrapolation procedure to be used if a specified value for the angle-of-attack falls beyond the last table value for the independent variable. A value of 1 indicates that the independent variable table value closest to the specified value will be used. The purpose of the next number is similar to that of the previous number in that it is an integer code for the extrapolation procedure to be used if a specified value for the angle-of-attack falls before the first table value for the independent variable. The next value of 1, in this case, indicates that the extrapolation is to be linear using the first two table values for the independent variable. Following these two integer codes are the nine values of the independent variable, angle-of-attack. IND3 has 4 Mach number values associated with it.

Ordered position has been indicated on the IND4 line by using a 0 for the format data-type code. The next value, 2, indicates there are two dependent variables in this table. The extrapolation procedure values are irrelevant in this line so they have been given values of 0. From examining the dependent data, the lift coefficients are listed first followed by the drag coefficients.

After all of the independent variables have been defined, the dependent variable is defined following the symbol DEP. As with the independent variable definitions, the character following the DEP symbol is a format data-type code. Once again, RS indicates the values of the lift coefficient are to be read in as real single precision numbers. Following this character are the values of the dependent variable.

It is important to place the dependent variables in the correct order when working with more than one independent variable. In this example, ANOPP will read the order position (IND4) first, Mach number (IND3) second, angle-of-attack (IND2) third, and spanwise station (IND1) fourth. If the "do loop" is used to visualize the order of the three independent variables, then the most inner do-loop is IND1, the next one is IND2, and the most outer one is IND3. Because of the presence of 0 in IND4 card and the number two after 0 shows that there are two ordered positions, this can become the most outer do-loop for the lift coefficients and drag coefficients.

The END\* symbol is the input terminator card which signifies the end of a table input section. This is also a control statement which requires a dollar sign (\$) at the end of the line. The statements between the line beginning with TABLE and the line beginning



with END\* are table description cards, not control statements, therefore, they do not require dollar sign symbols (\$) at the end of each line.

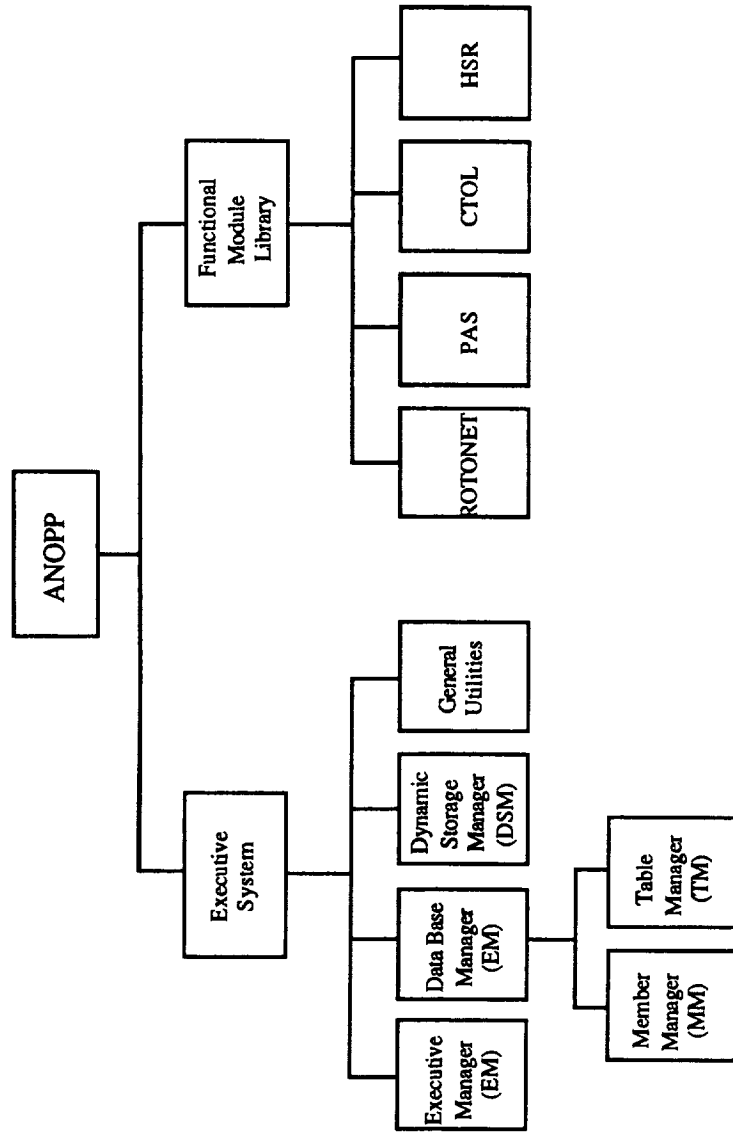


Figure 1. ANOPP Functional Hierarchy

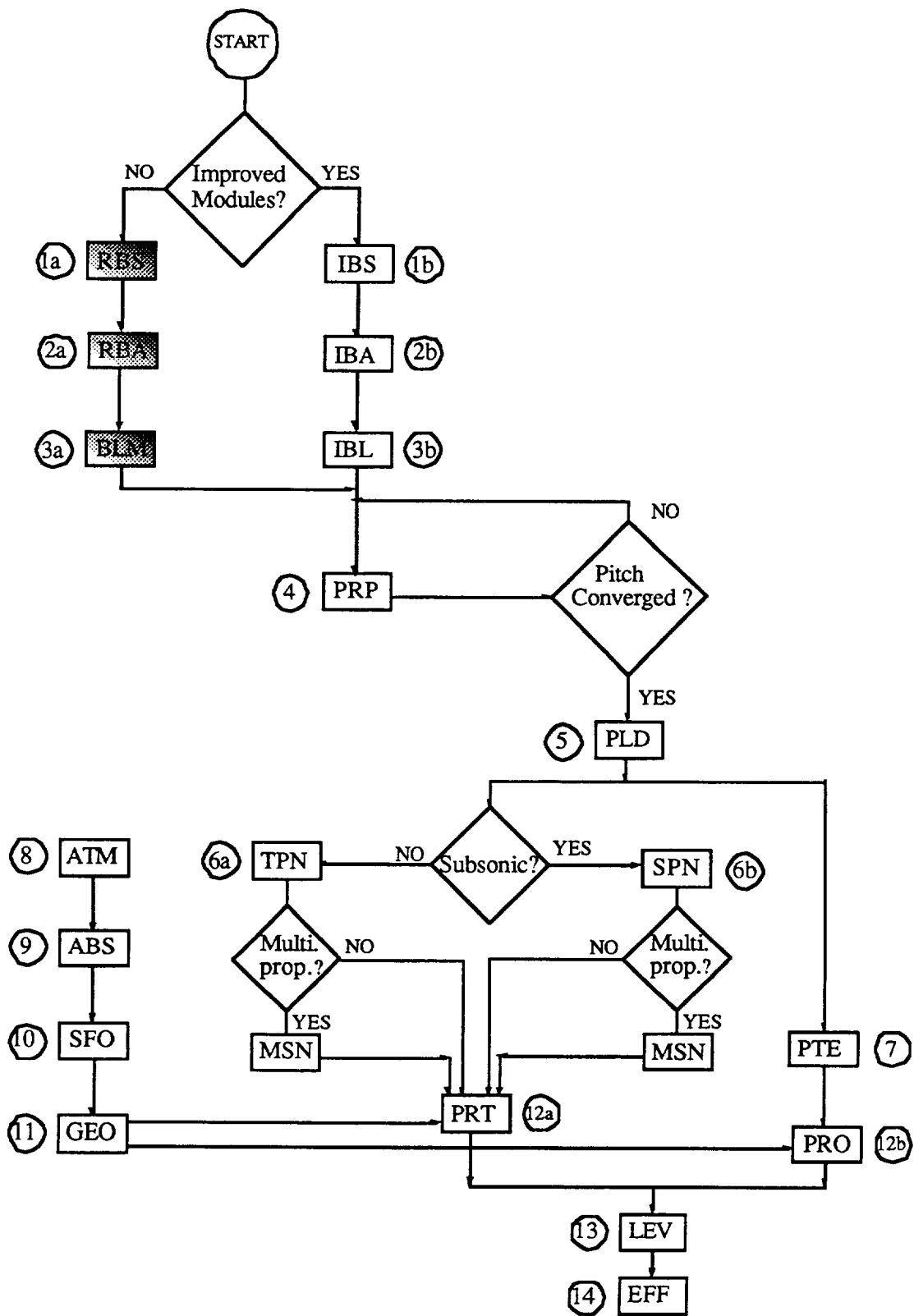


Figure 2. Flowchart of ANOPP-PAS program modules used for wind tunnel and flight predictions

```

* * *
*   PURPOSE - short description of the functional module
*
*   AUTHOR - programmer's initials and level number, such as L01/00/00
*
*   INPUT
*       USER PARAMETERS
*           Name1 - description
*
*           .
*
*           .
*
*           Namen - description
*
*   REAL USER PARAMETER LIMITS - SI UNITS
*   PARAMETER      MINIMUM  MAXIMUM  DEFAULT
*   Name1          number    number    number
*
*   .
*
*   .
*
*   Namen          number    number    number
*
*   REAL USER PARAMETER LIMITS - ENGLISH UNITS
*   PARAMETER      MINIMUM  MAXIMUM  DEFAULT
*   Name1          number    number    number
*
*   .
*
*   .
*
*   Namen          number    number    number
*
*   INTEGER/LOGICAL/ALPHA PARAMETER UNITS
*   PARAMETER      MINIMUM  MAXIMUM  DEFAULT
*   Name1          number    number    number
*
*   .
*
*   .
*
*   Namen          number    number    number
*
*   MEMBERS
*       DATA UNIT(DATA MEMBER)
*
*   TEMPORARIES
*   MEMBERS
*       DATA UNIT(DATA MEMBER)
*
*   OUTPUT
*       SYSTEM PARAMETERS
*           Name - description
*       USER PARAMETERS - same as for INPUT
*       MEMBERS
*           DATA UNIT(DATA MEMBER)

```

```

* DATA BASE STRUCTURES
*   DATA UNIT(DATA MEMBER) - complete description of data and required
*                             format for all input and output data units
*
* ERRORS
*   NON-FATAL - description of errors that are possible within the
*               functional module.
*   FATAL -    functional modules do not use fatal errors.
*
* LDS REQUIREMENTS - describes the amount of local dynamic storage
*                   required by this module.
*
* GDS REQUIREMENTS - describes the amount of global dynamic storage
*                   required for this module.
*
* * *

```

Figure 3. - ANOPP functional module prologue format

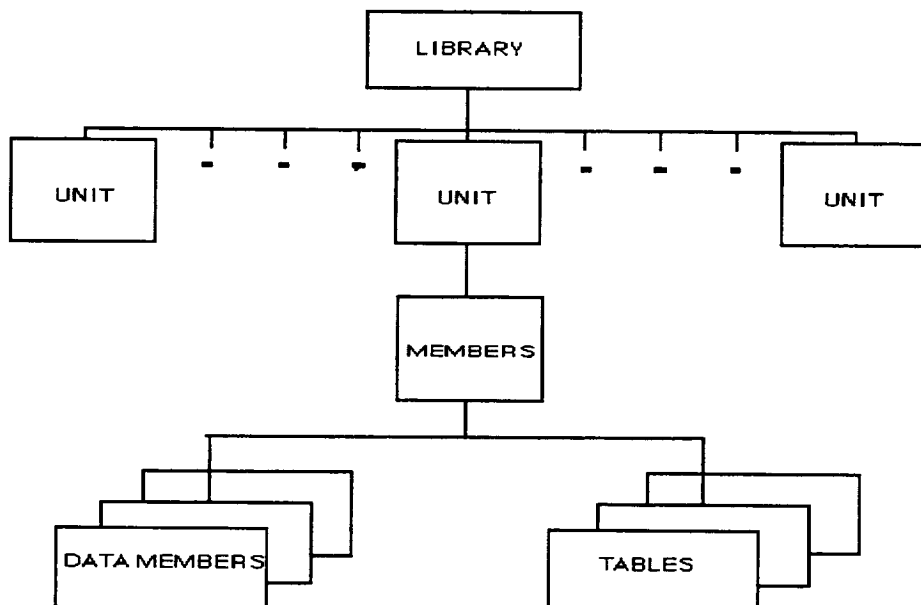


Figure 4. Library Hierarchy

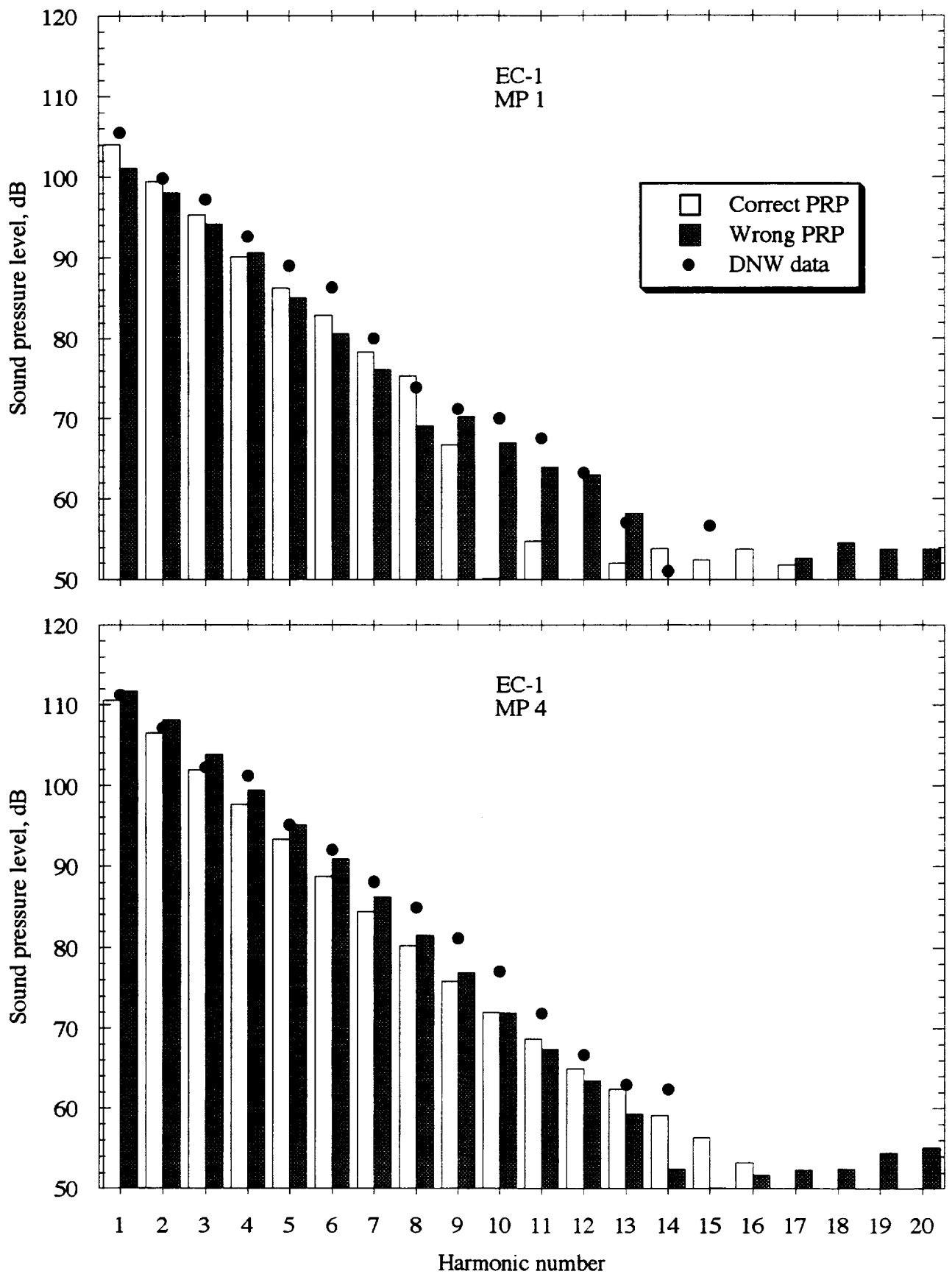


Fig. 5 Comparison of results of incorrect and modified equations in PRP with DNW data.

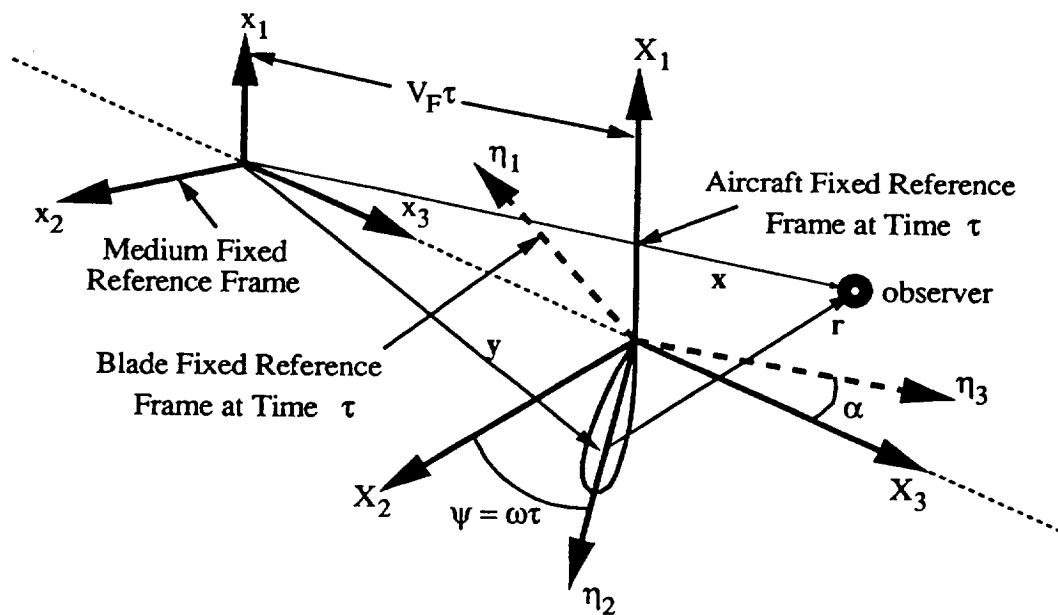


Fig. 6 Reference frames.

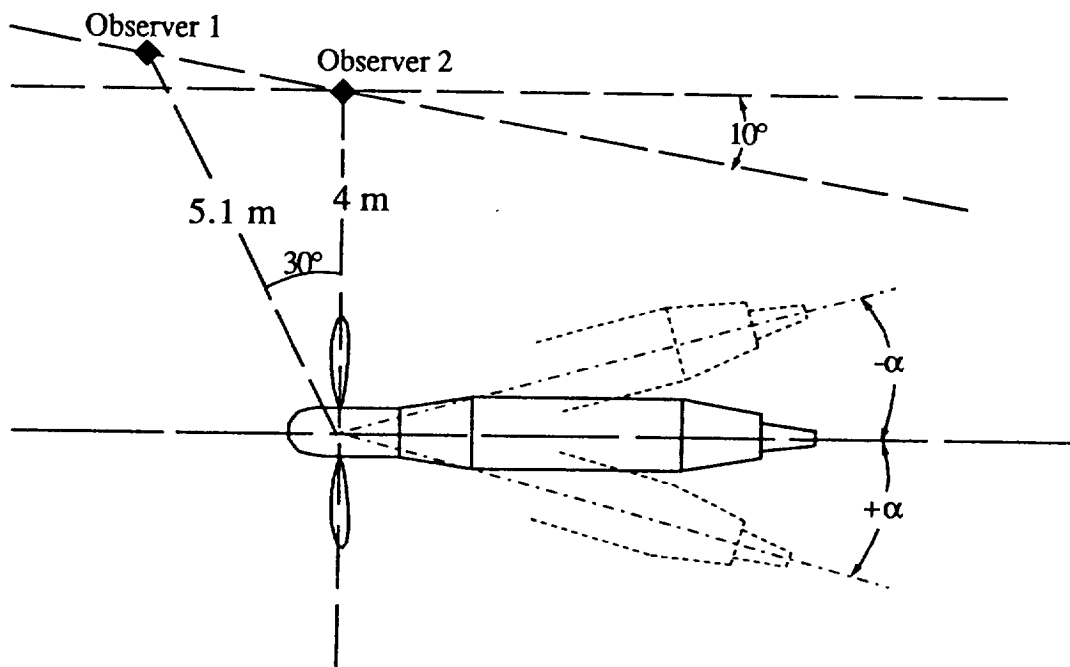


Fig. 7 Microphone position relative to propeller in the DNW test.

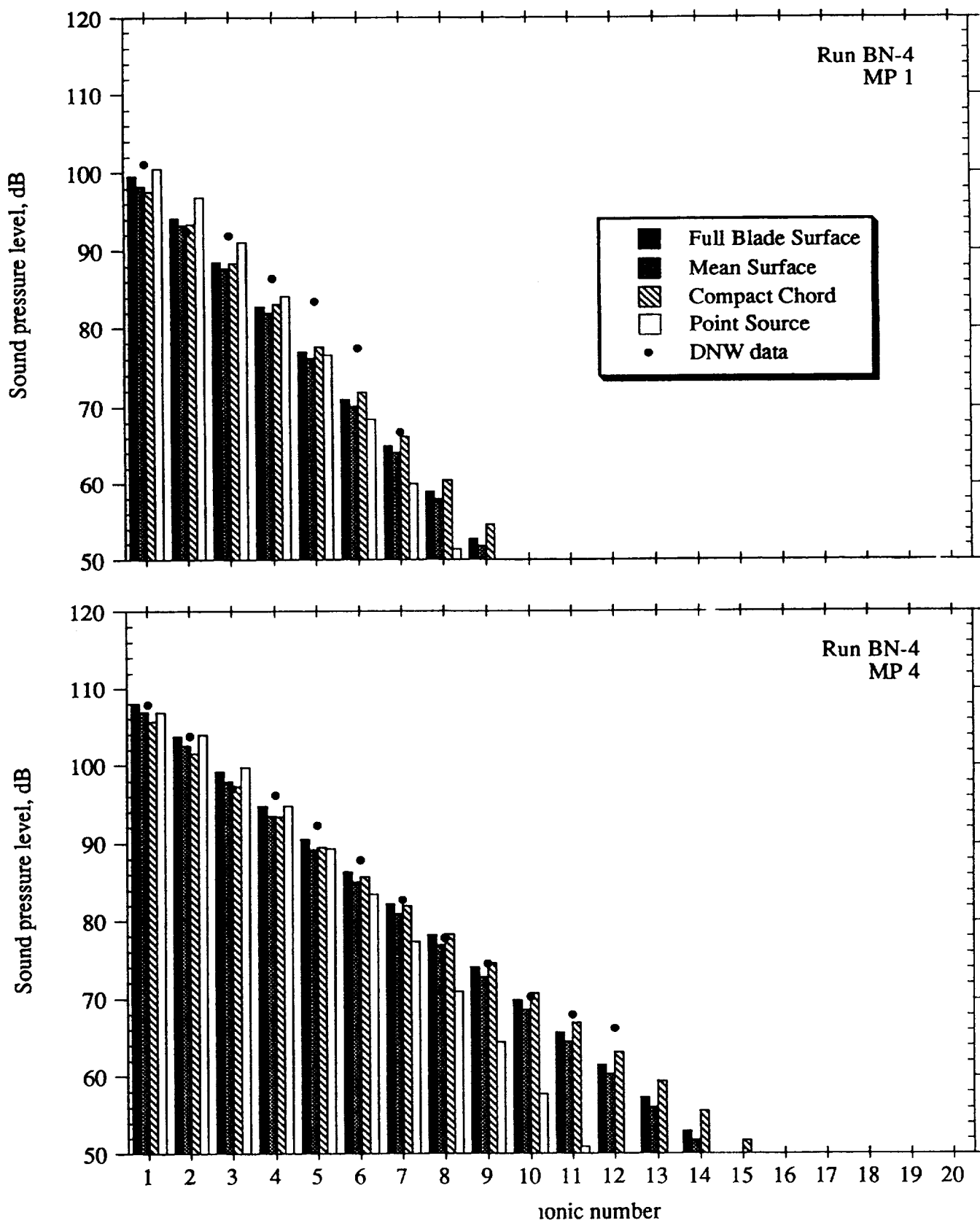


Fig. 8 Comparison of the predictions of the four methods and DNW data.  
 $\alpha = 0^\circ$



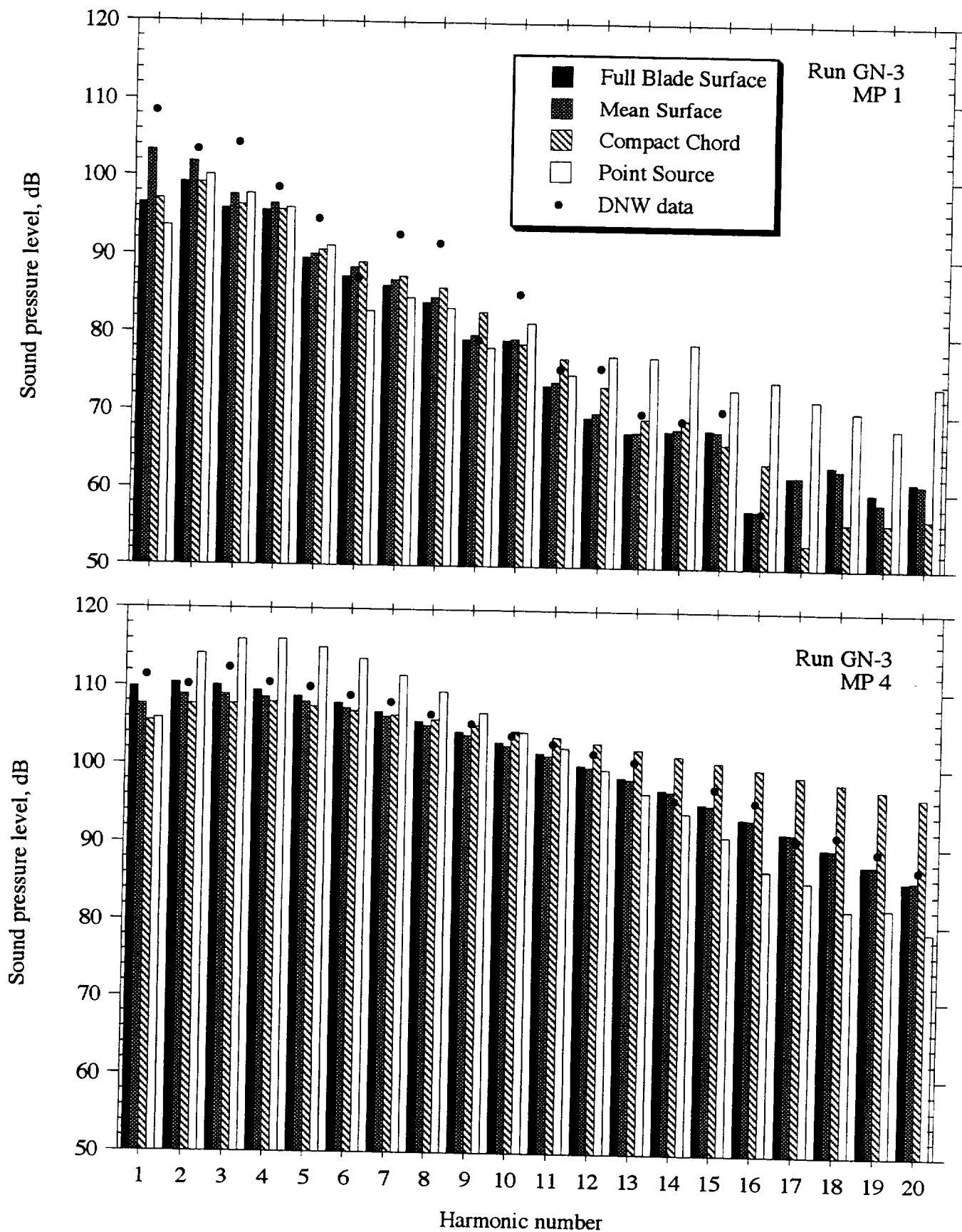


Fig. 9 Comparison of the predictions of the four methods and DNW data.  
 $\alpha = -7.4^\circ$

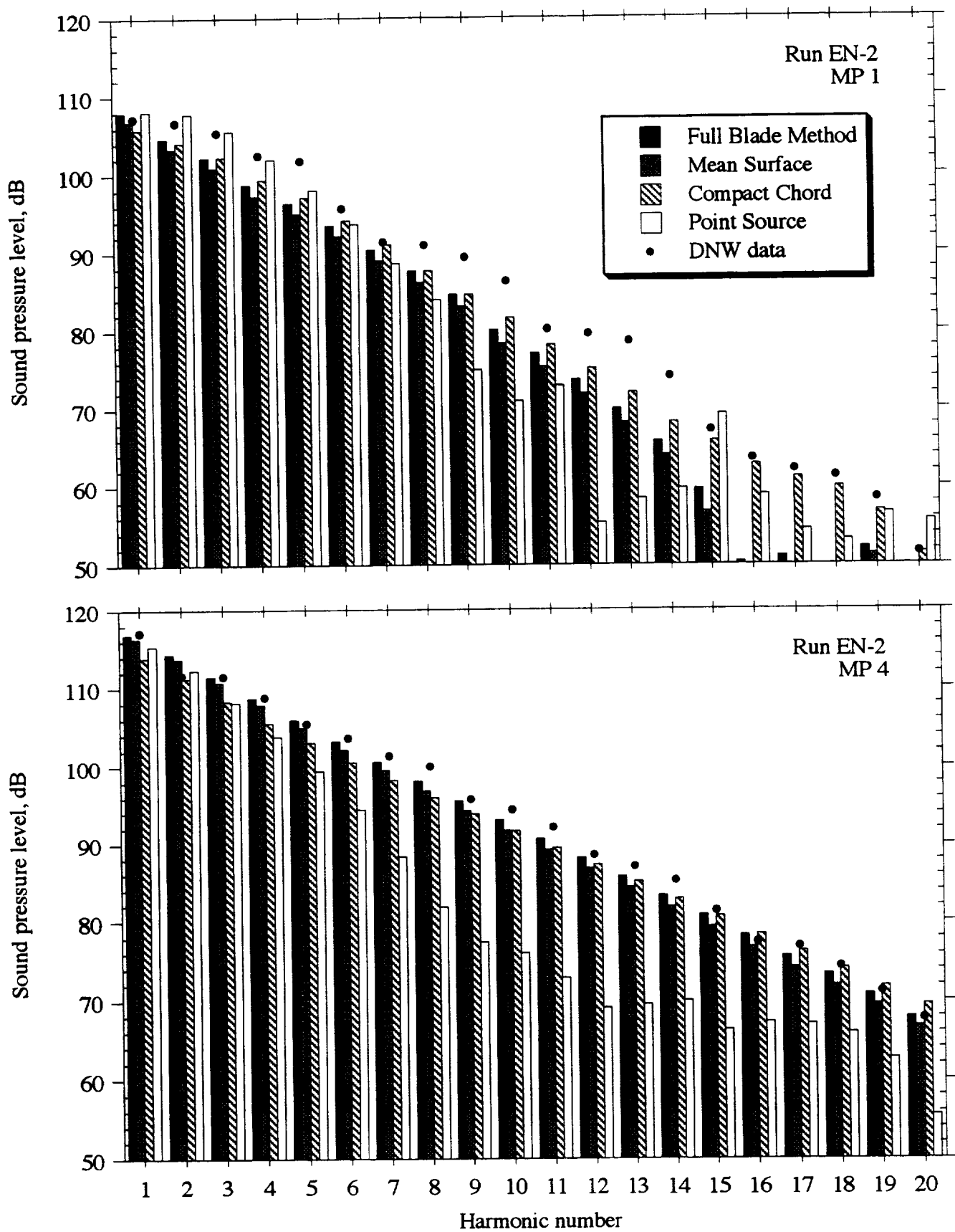


Fig. 10 Comparison of the predictions of the four methods and DNW data.  
 $\alpha = 7.3^\circ$

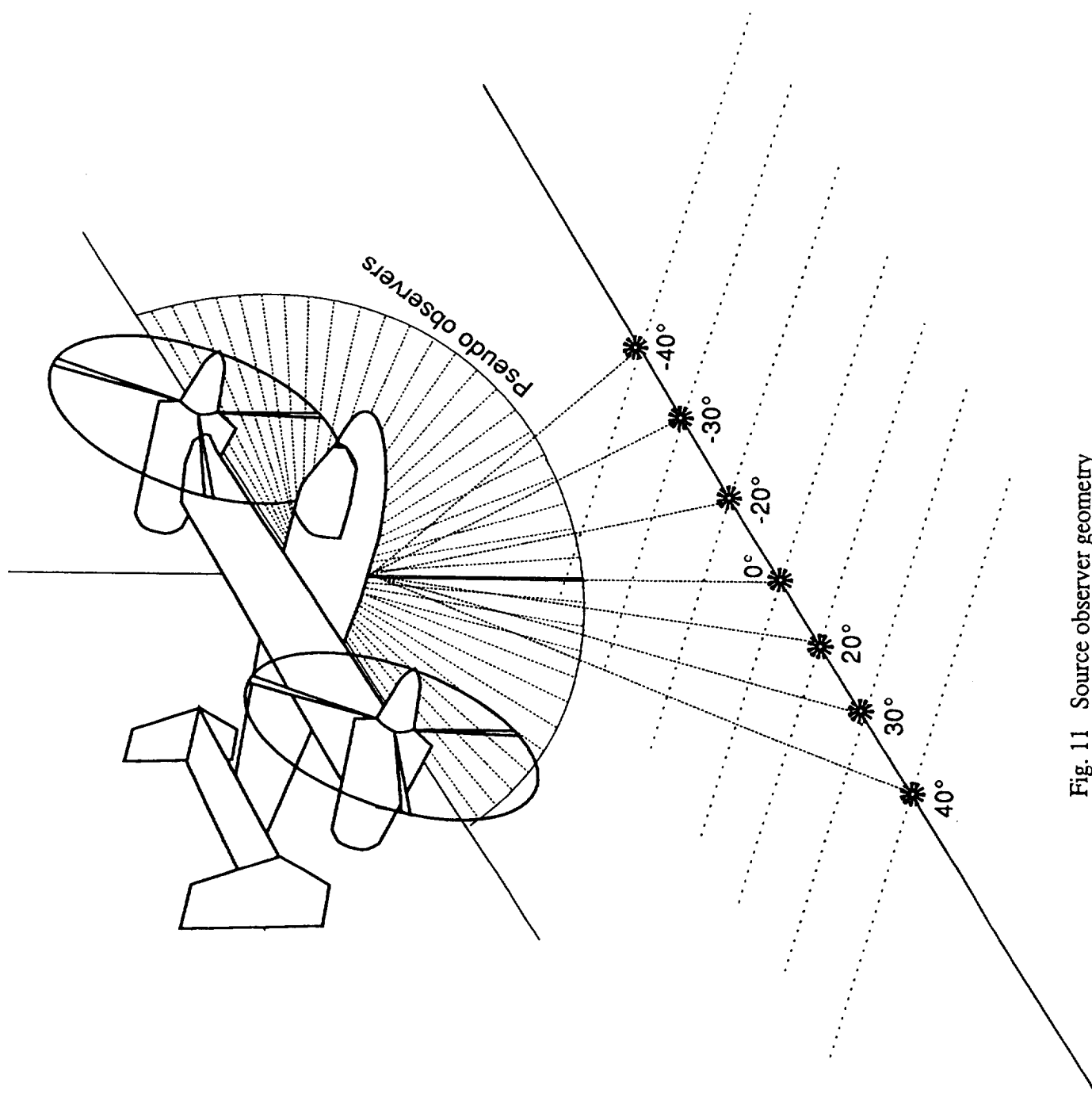
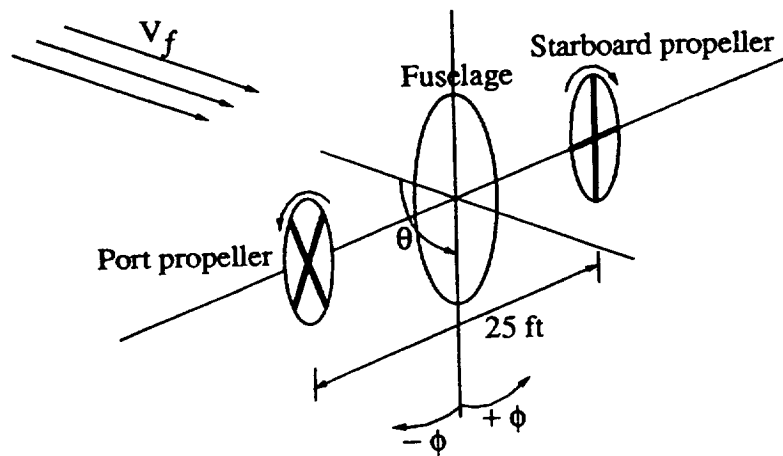


Fig. 11 Source observer geometry

Case 1



Case 2

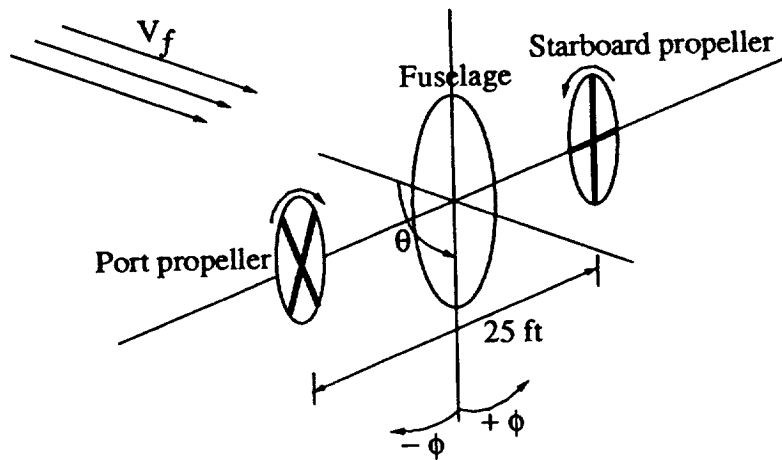


Fig. 12 Geometries of the case studies

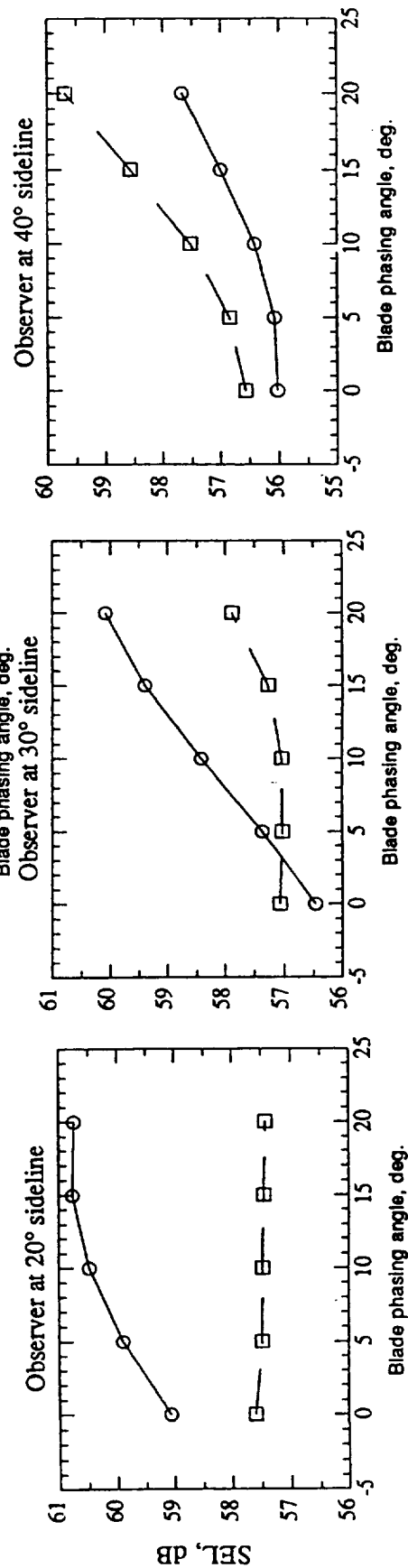
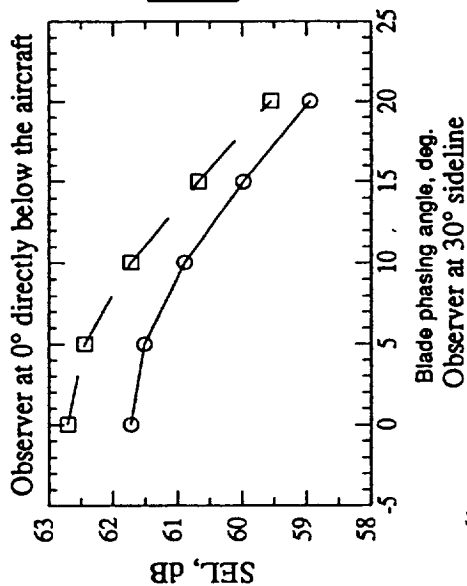
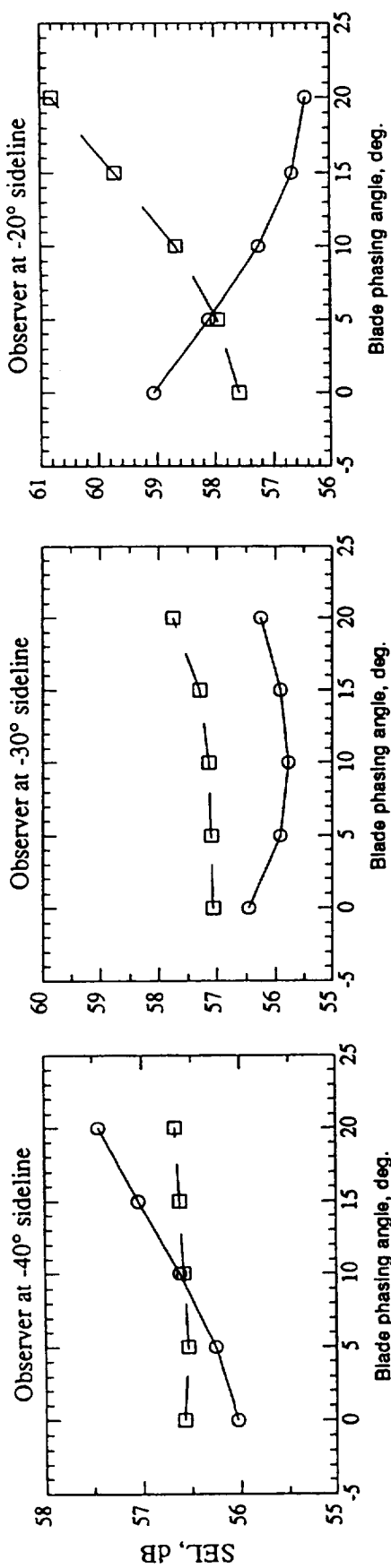


Fig. 13 Effect of blade phasing angle on SEL for propellers turning away from each other (case 1) and toward each other (case 2).

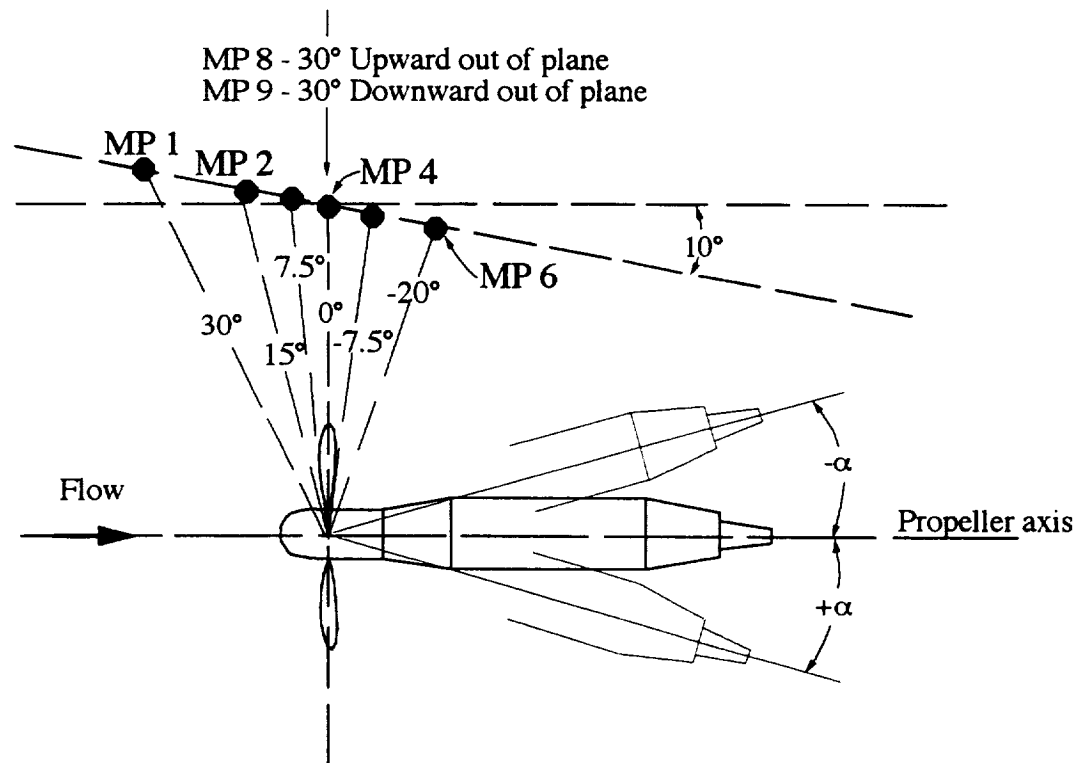


Fig. 14 Microphone position relative to propeller in the DNW experiment.

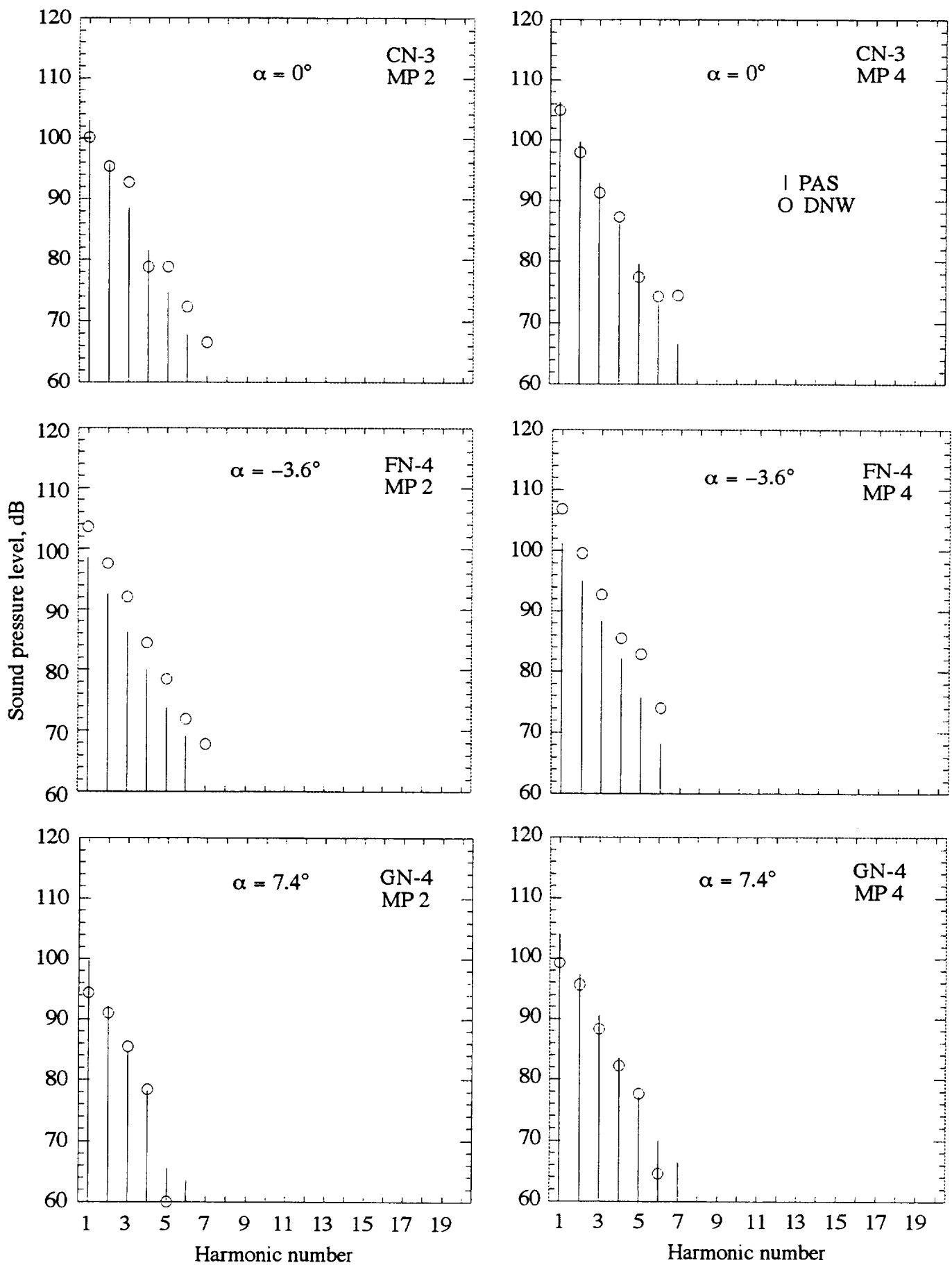


Fig. 15 Comparison of PAS prediction with DNW data at MP2 and MP4 for the round tip propeller.  $\Omega = 1800$  RPM

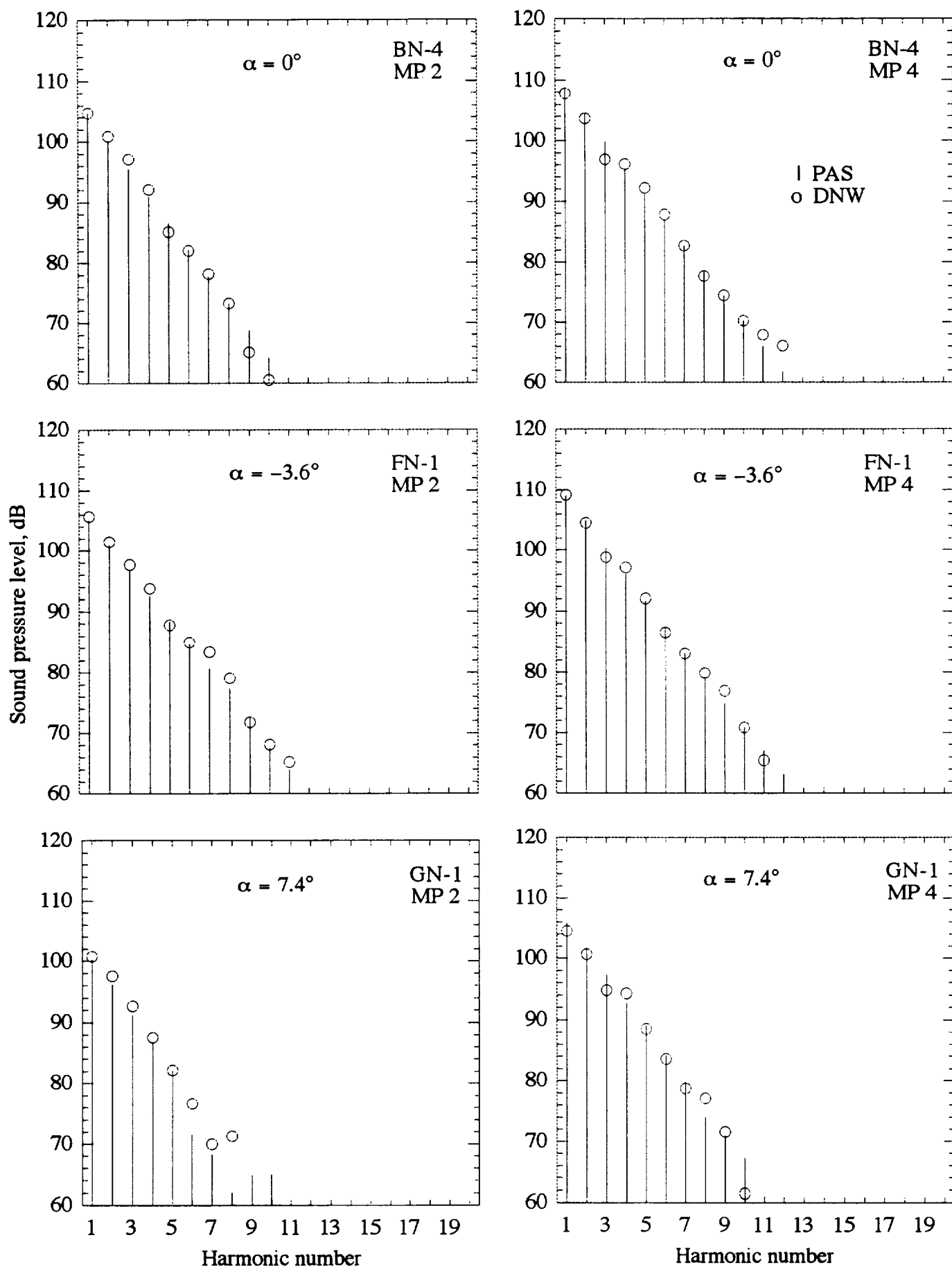


Fig. 16 Comparison of prediction with DNW data at MP 2 and MP 4 for the round tip propeller.  
 $\Omega = 2100$  RPM



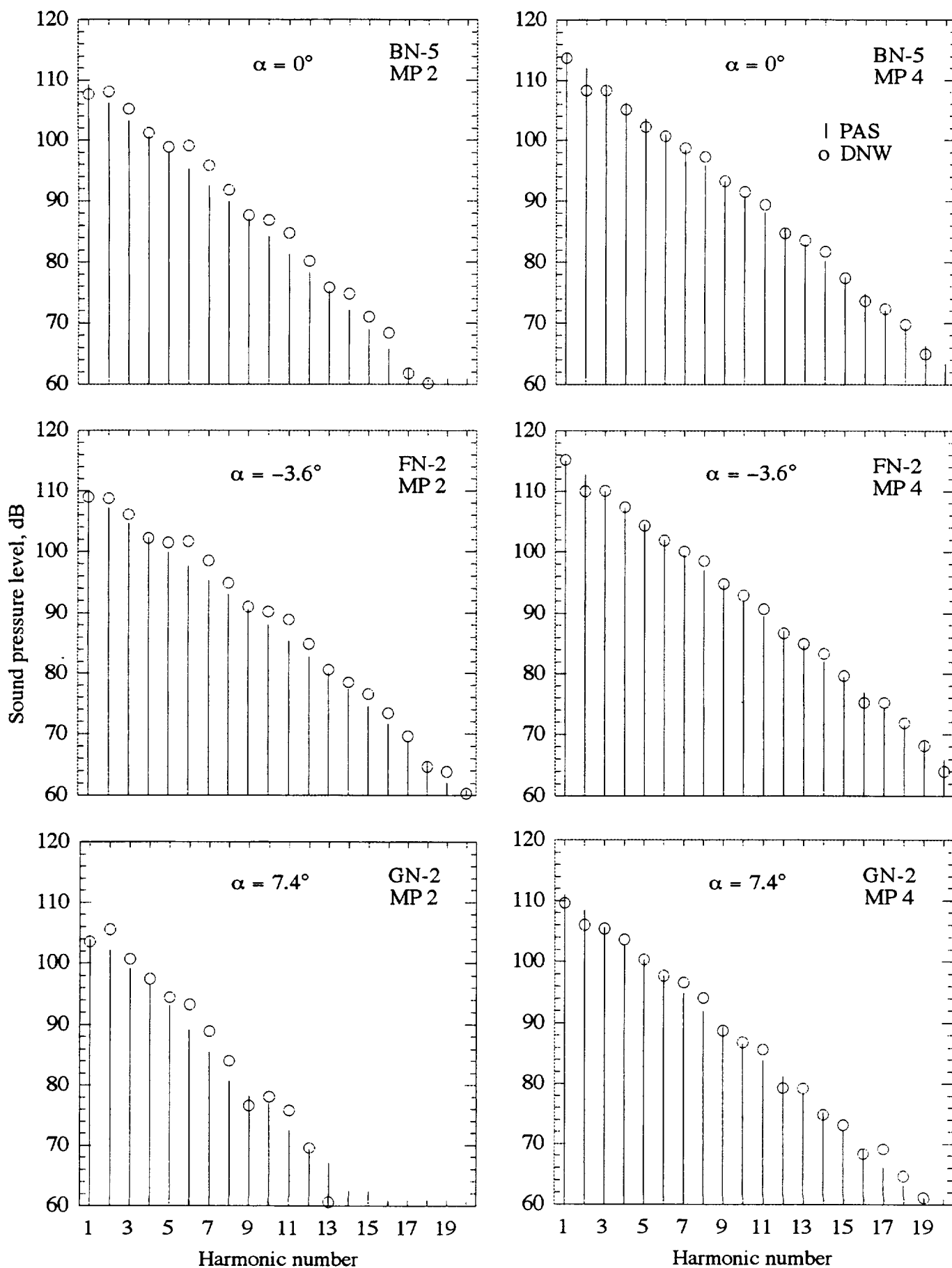


Fig. 17 Comparison of prediction with DNW data at MP 2 and MP4 for the round tip propeller.  
 $\Omega = 2400$

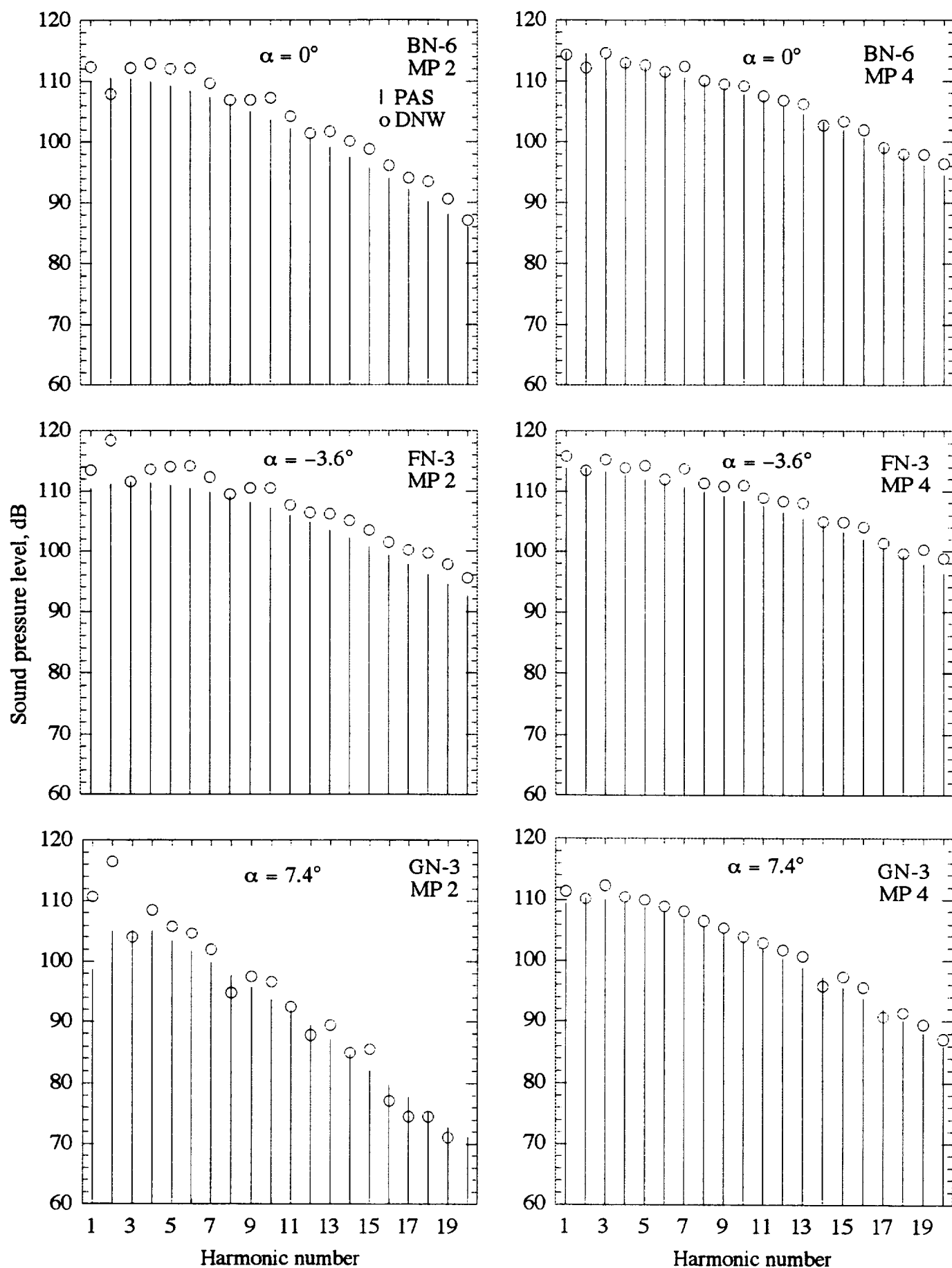


Fig. 18 Comparison of prediction with data at MP2 and MP4 for the round tip propeller.  
 $\Omega = 2700$

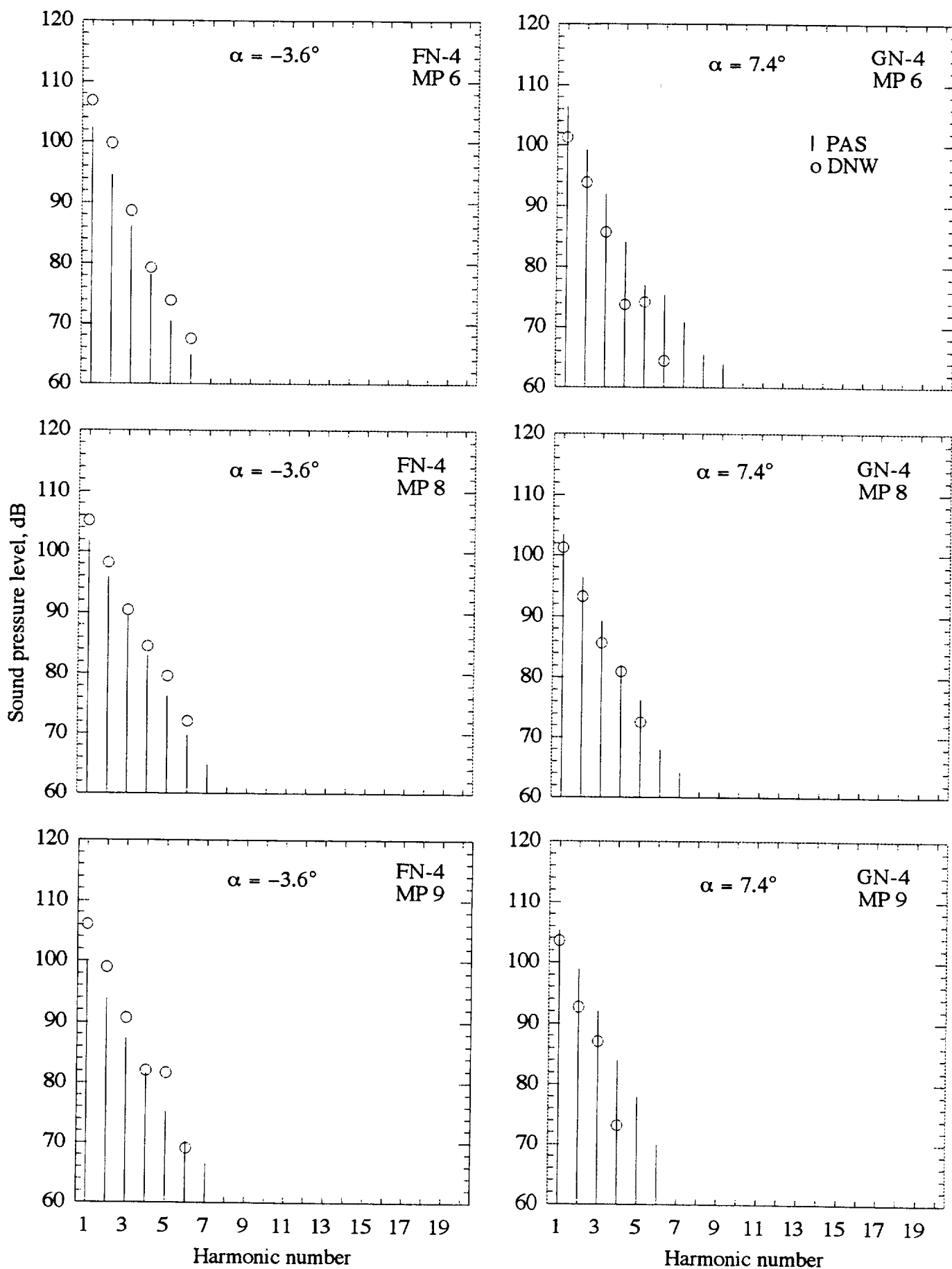


Fig. 19 Comparison of prediction with DNV data at MP6, MP8 and MP9 for the round tip propeller.  $\Omega = 1800$  RPM

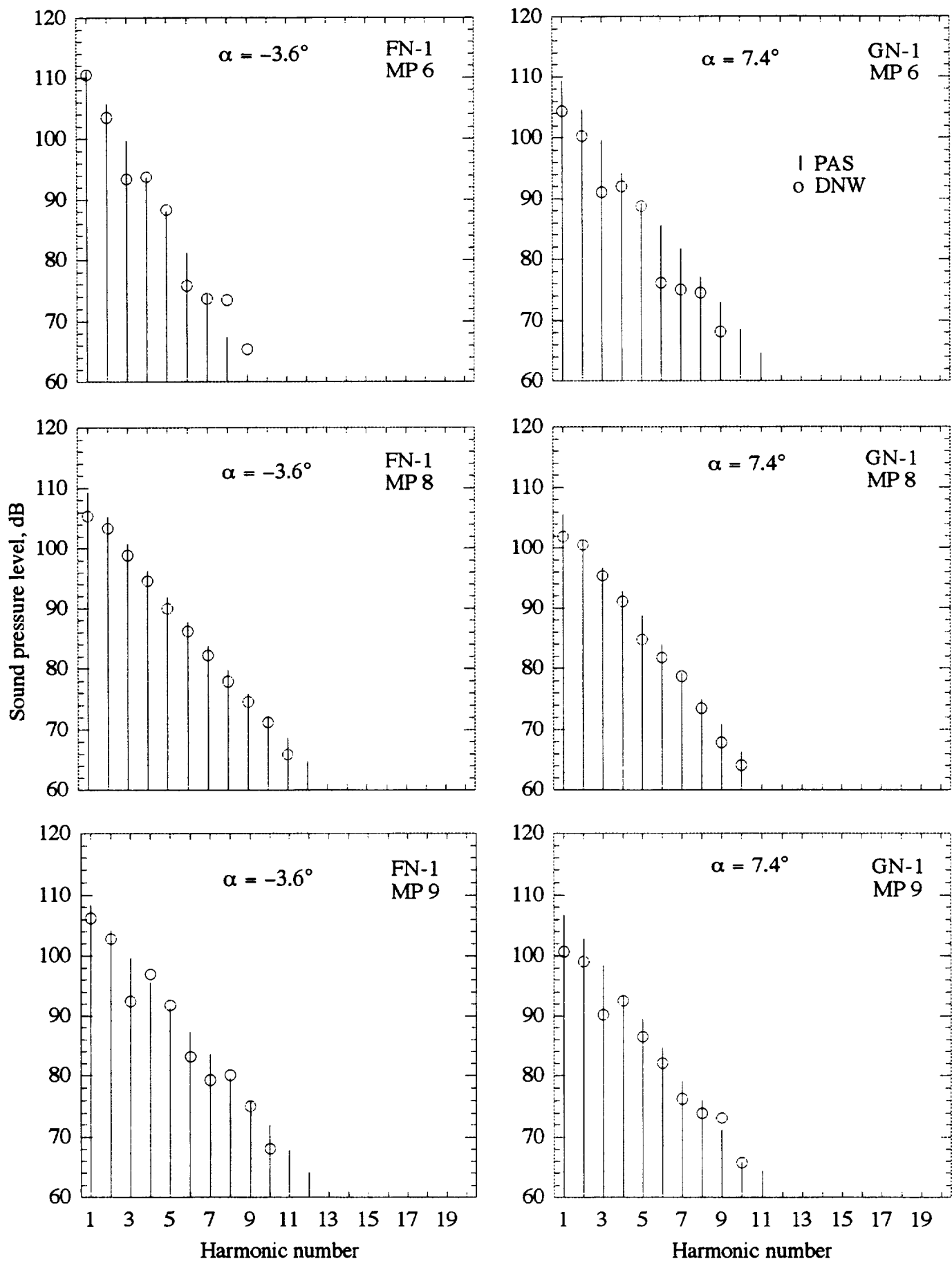


Fig. 20 Comparison of prediction with DNW data at MP6, MP8 and MP9 for the round tip propeller.  $\Omega = 2100$  RPM

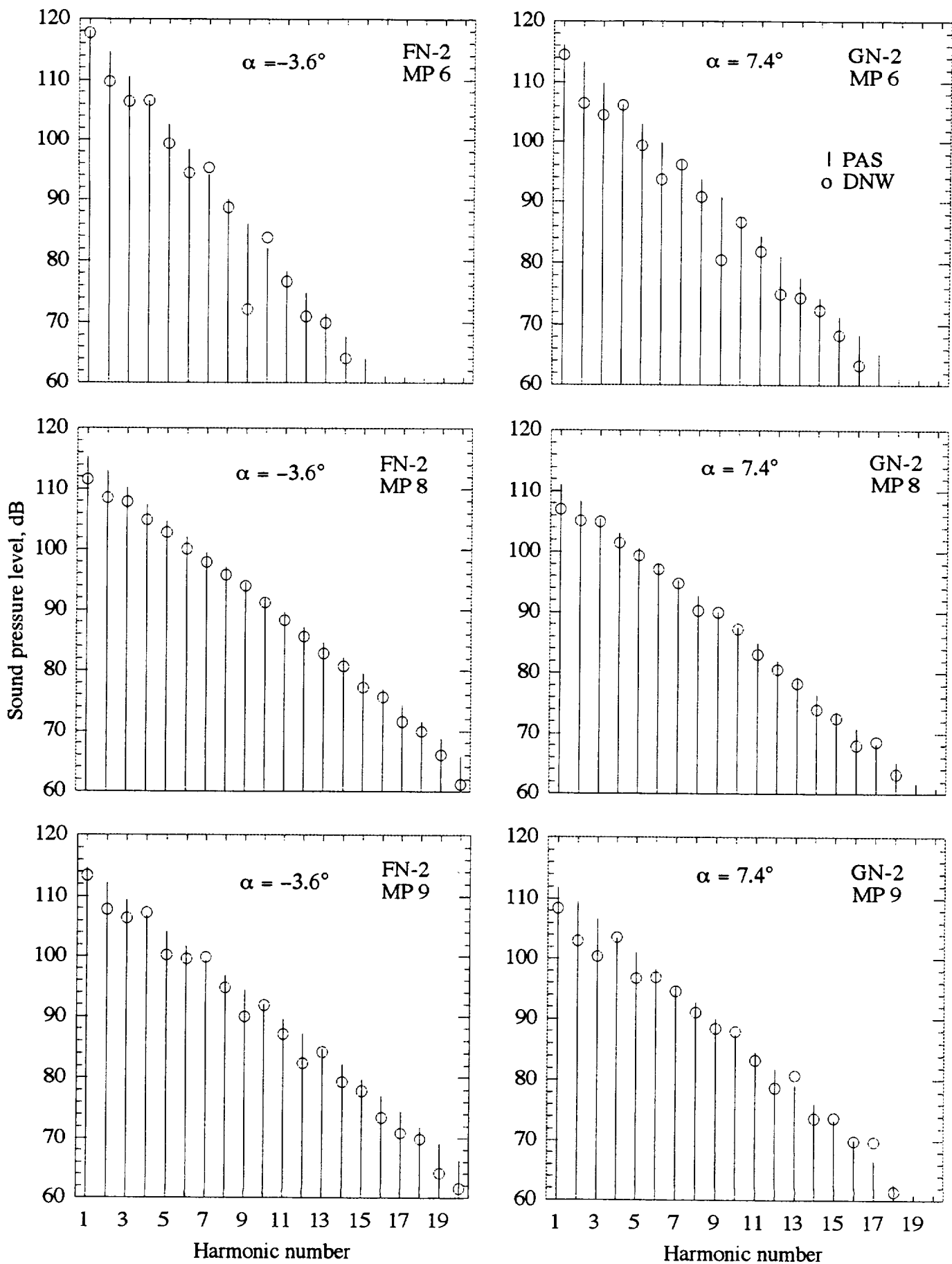


Fig. 21 Comparison of prediction with DNW data at MP6, MP8 and MP9 for the round tip propeller.  $\Omega = 2400$  RPM

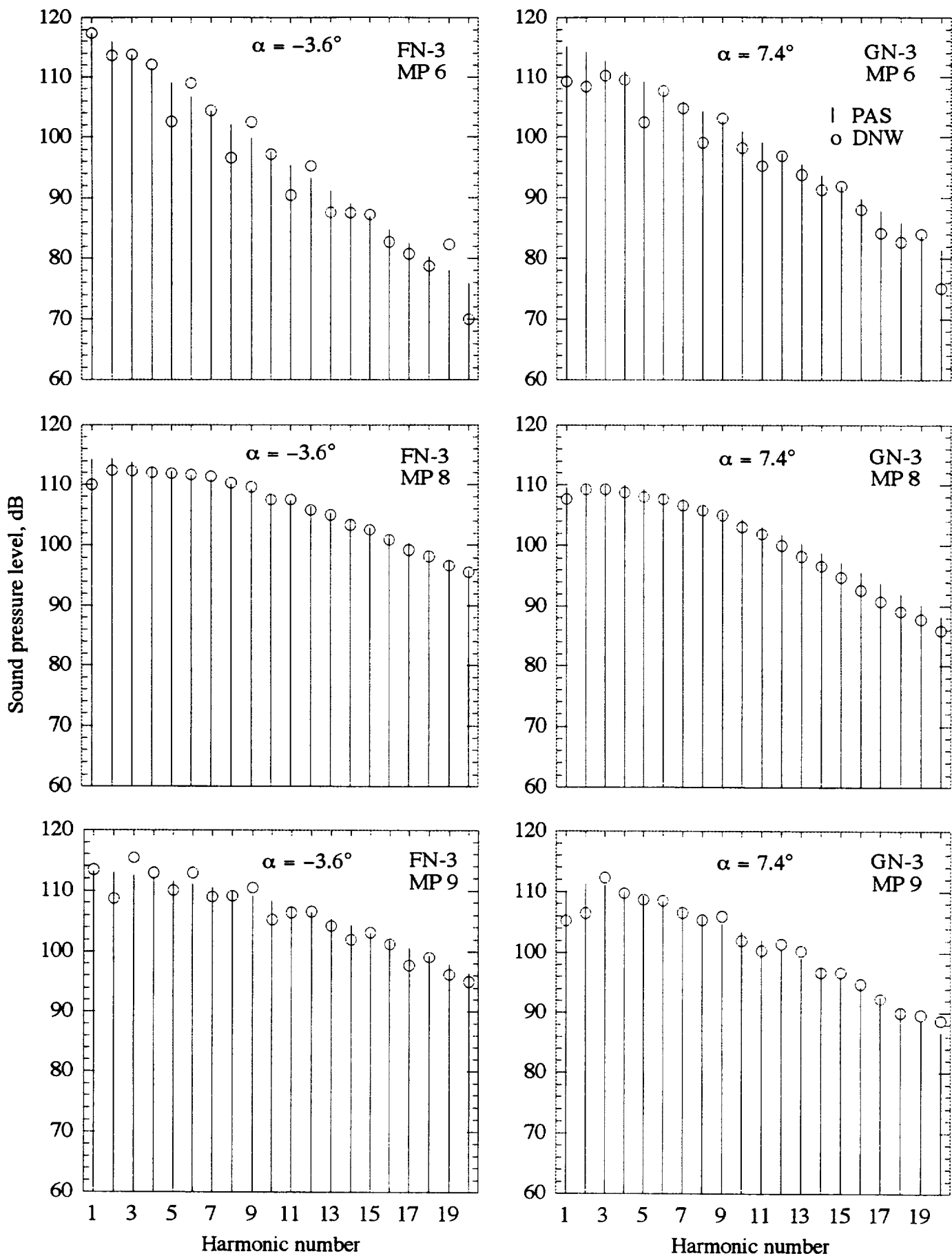


Fig. 22 Comparison of prediction with DNW data at MP6, MP8 and MP9 for the round tip propeller.  $\Omega = 2700$  RPM

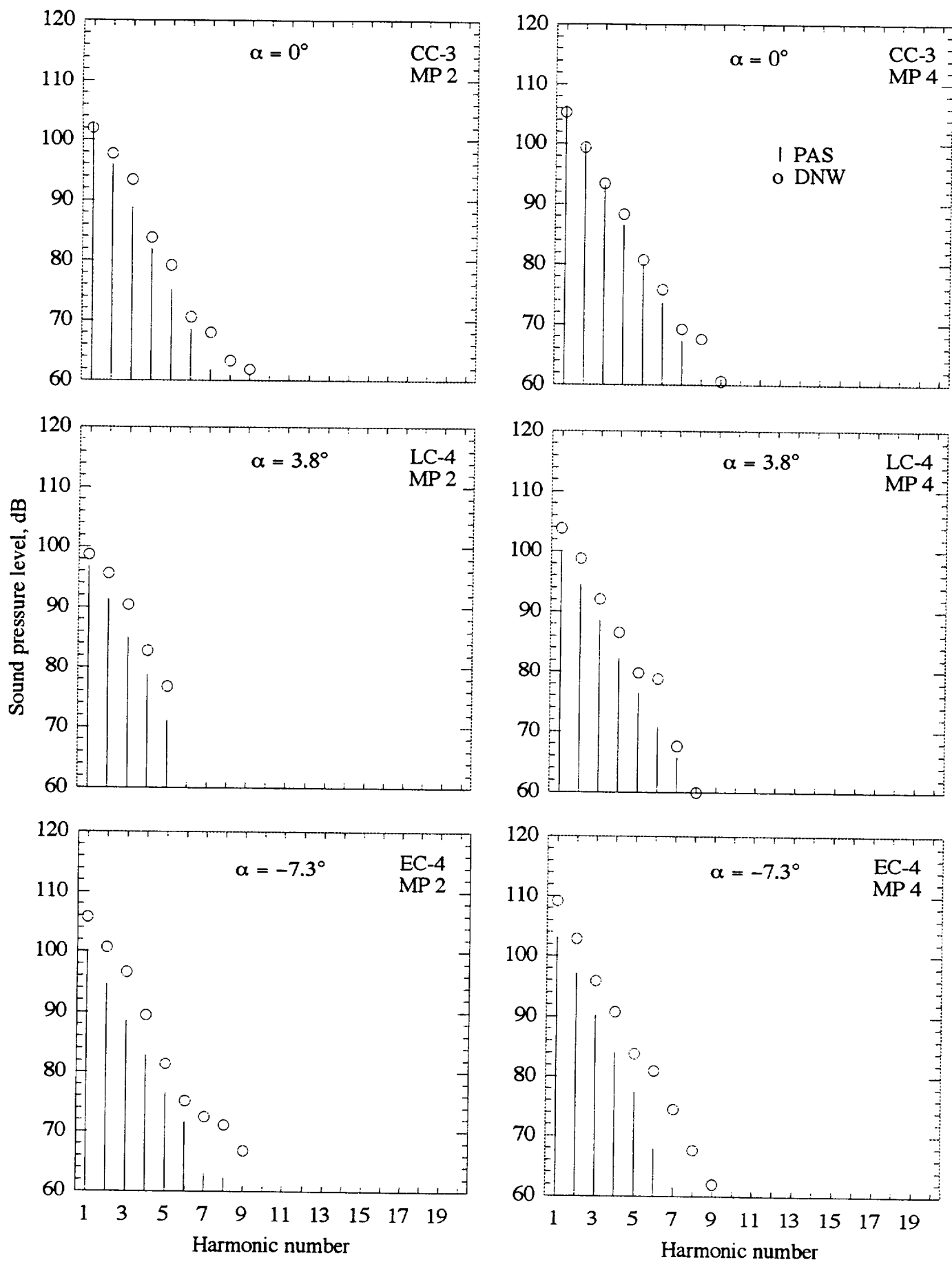


Fig. 23 Comparison of prediction with data at MP2 and MP4 for the square tip propeller.  
 $\Omega = 1800$

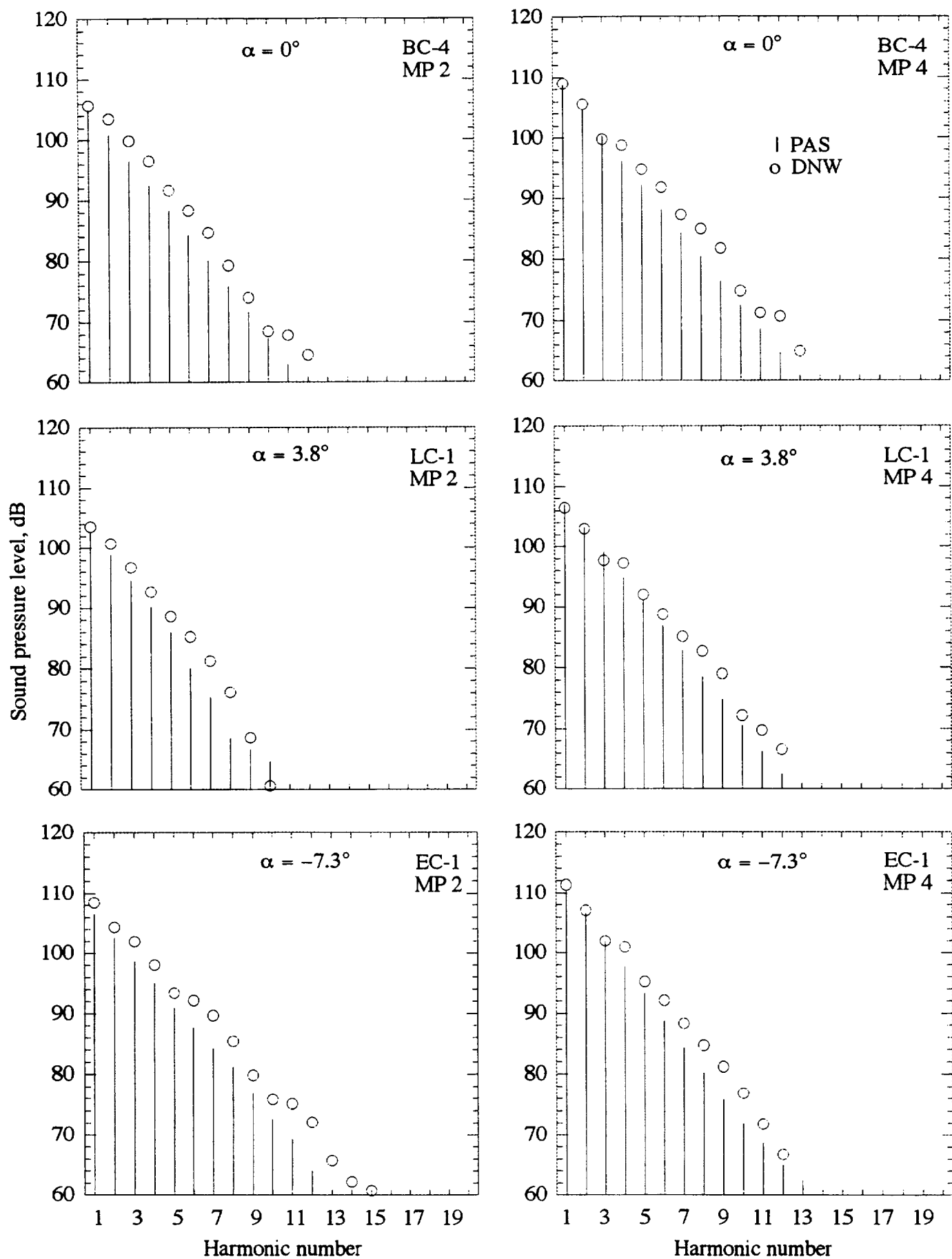


Fig. 24 Comparison of prediction with data at MP2 and MP4 for the square tip propeller.  
 $\Omega = 2100$



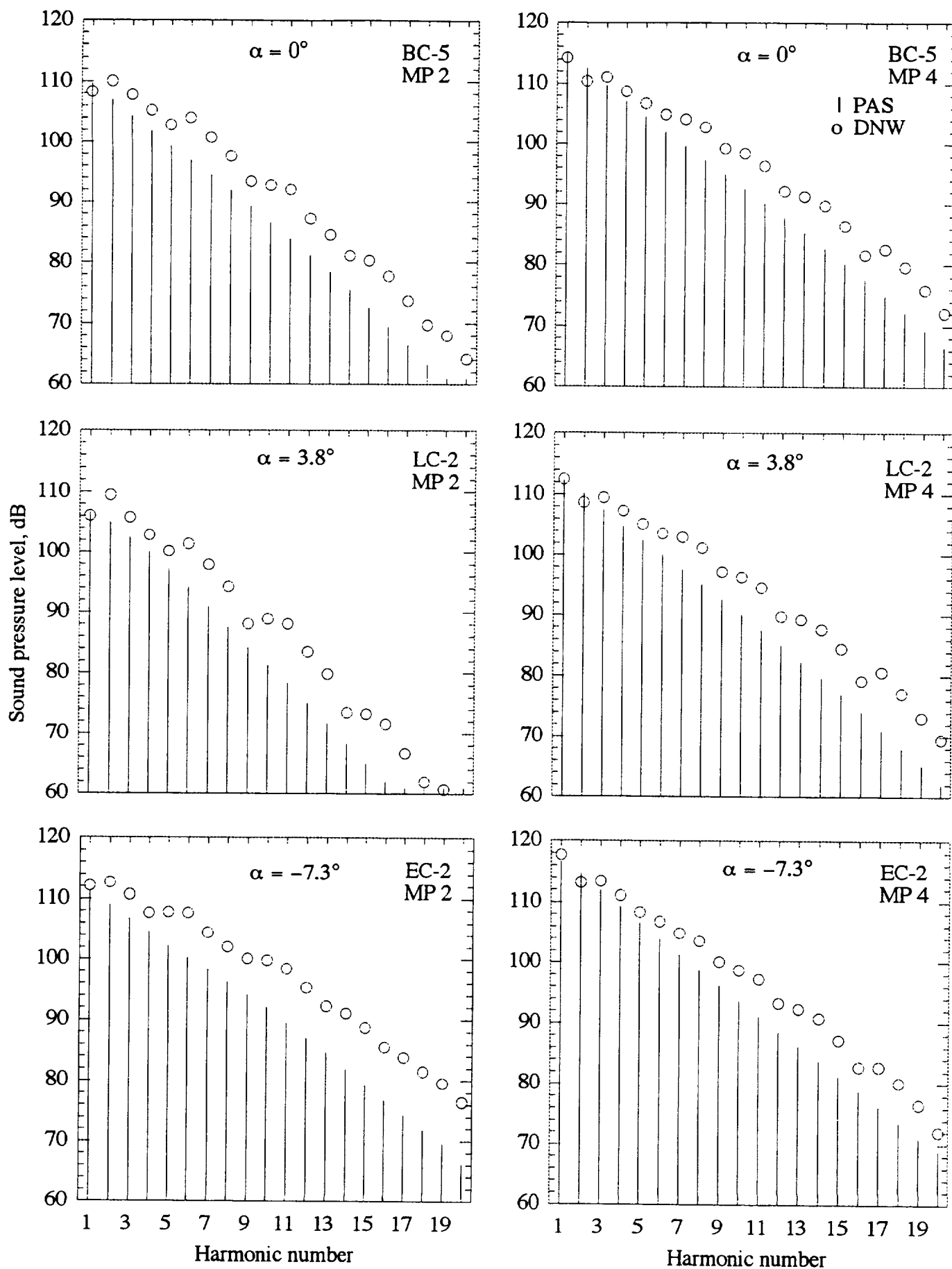


Fig. 25 Comparison of prediction with data at MP2 and MP4 for the square tip propeller.  
 $\Omega = 2400$

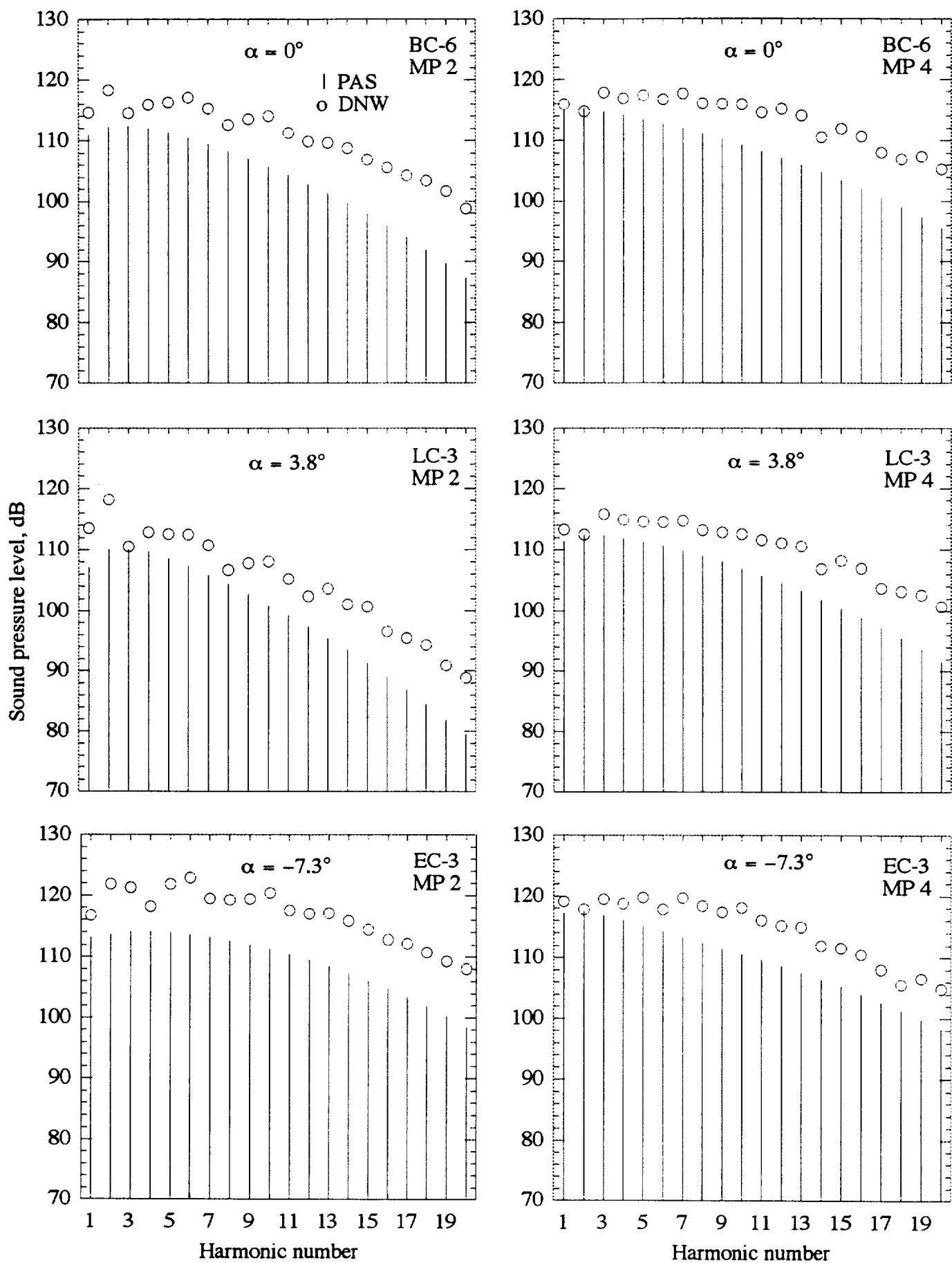


Fig. 26 Comparison of prediction with data at MP2 and MP4 for the square tip propeller.  
 $\Omega = 2700$

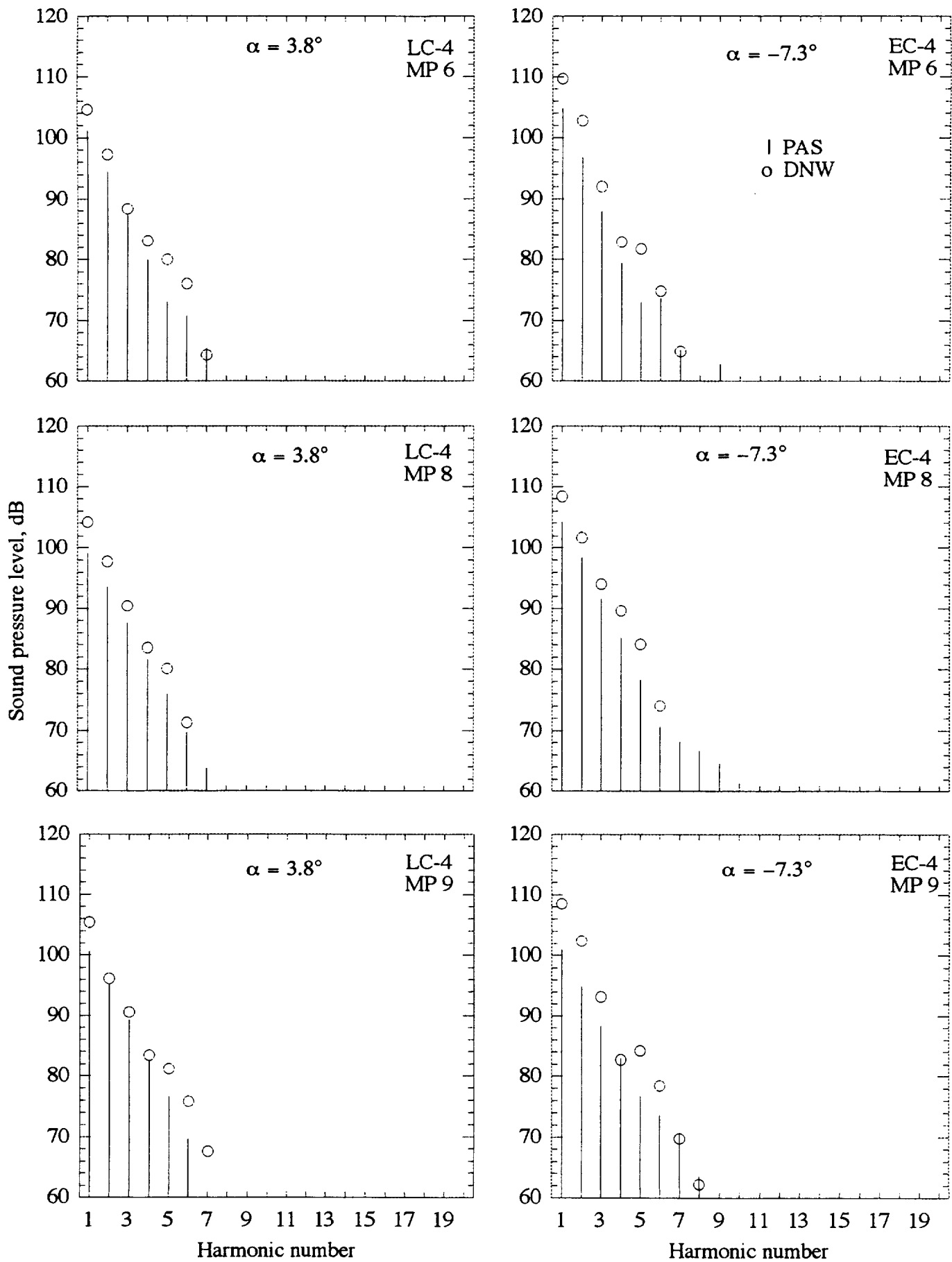


Fig. 27 Comparison of prediction with DNW data at MP6, MP8 and MP9 for the square tip propeller.  $\Omega = 1800$  RPM

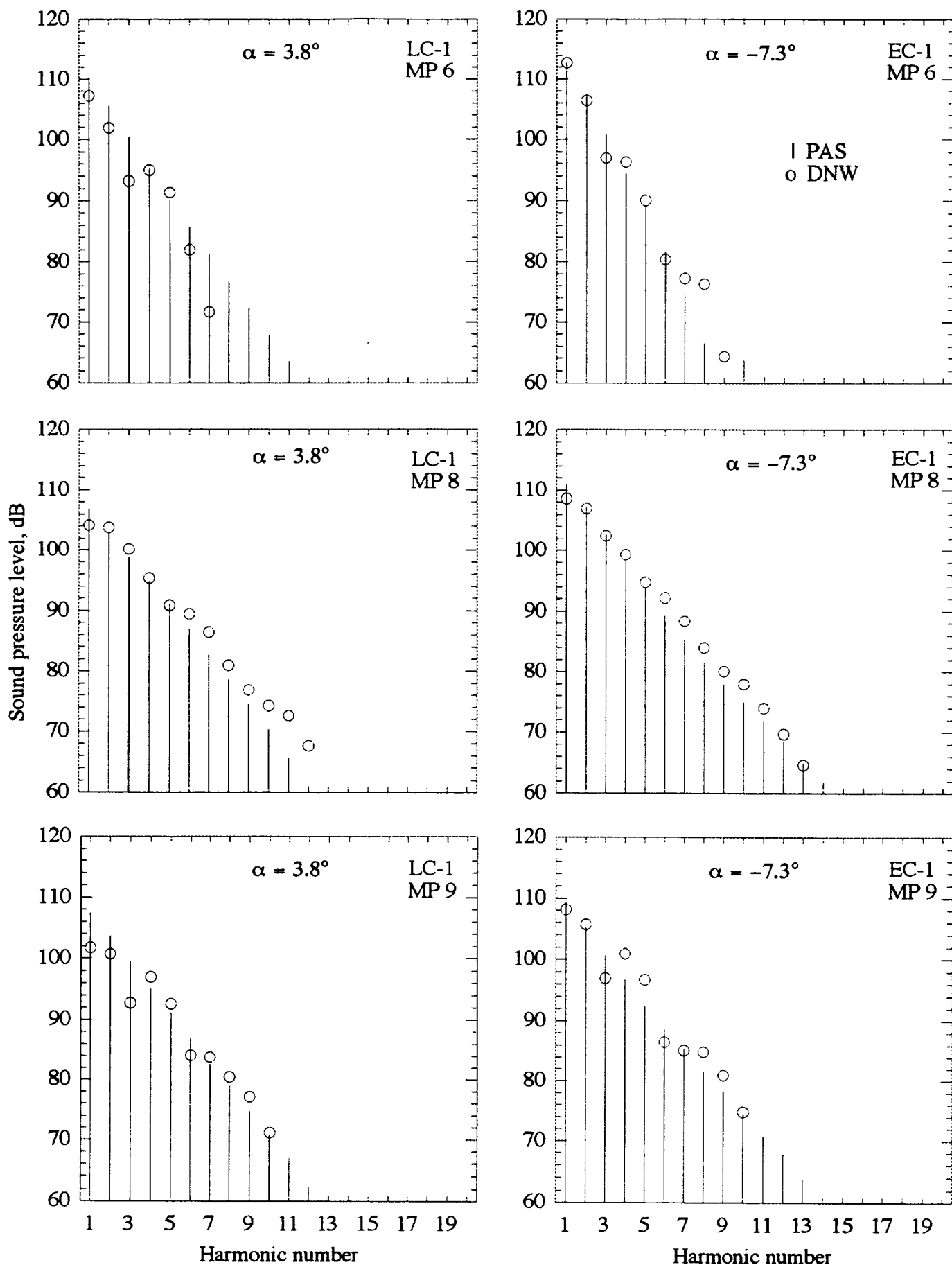
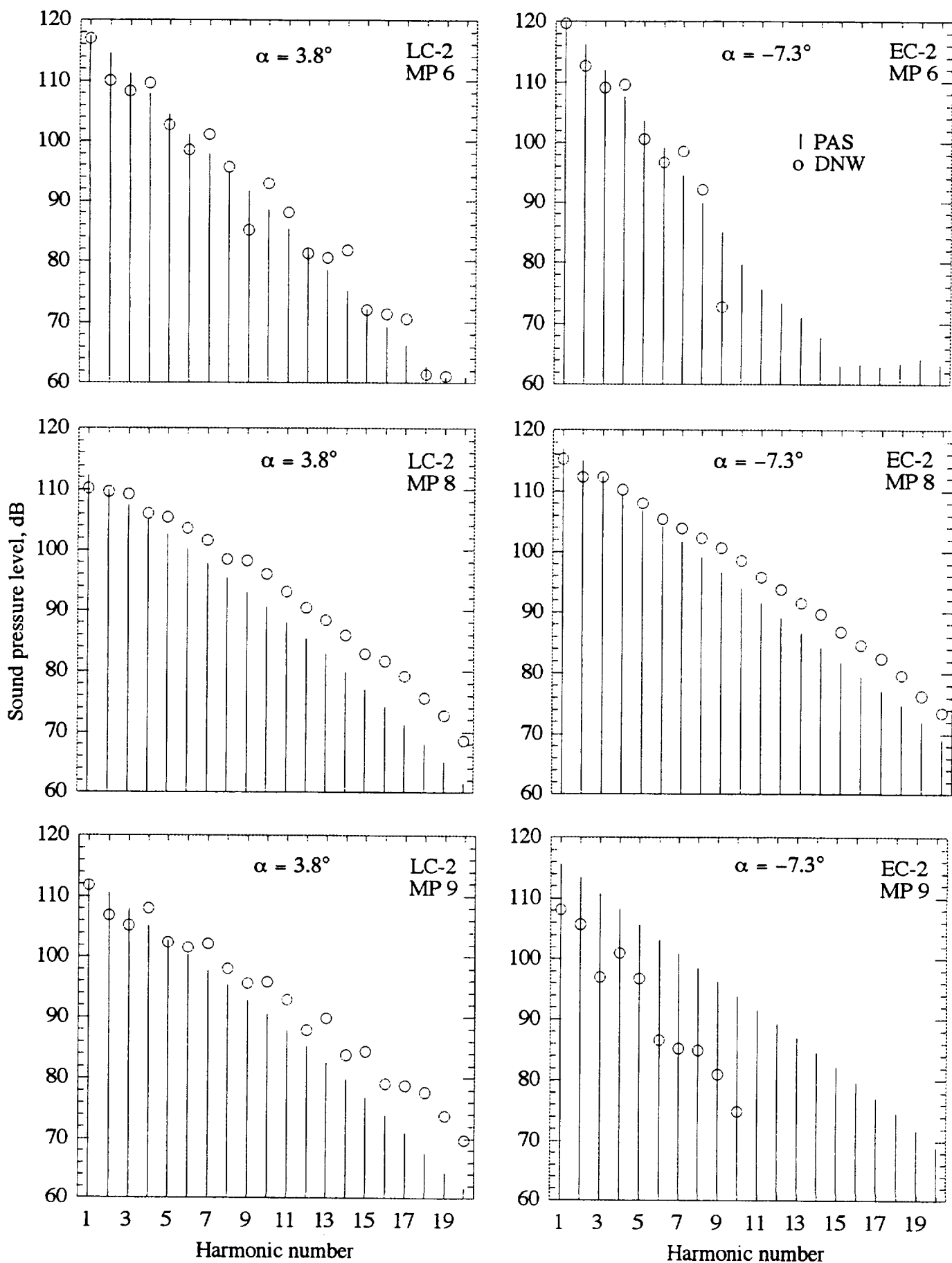


Fig. 28 Comparison of prediction with DNW data at MP6, MP8 and MP9 for the square tip propeller.  $\Omega = 2100$  RPM



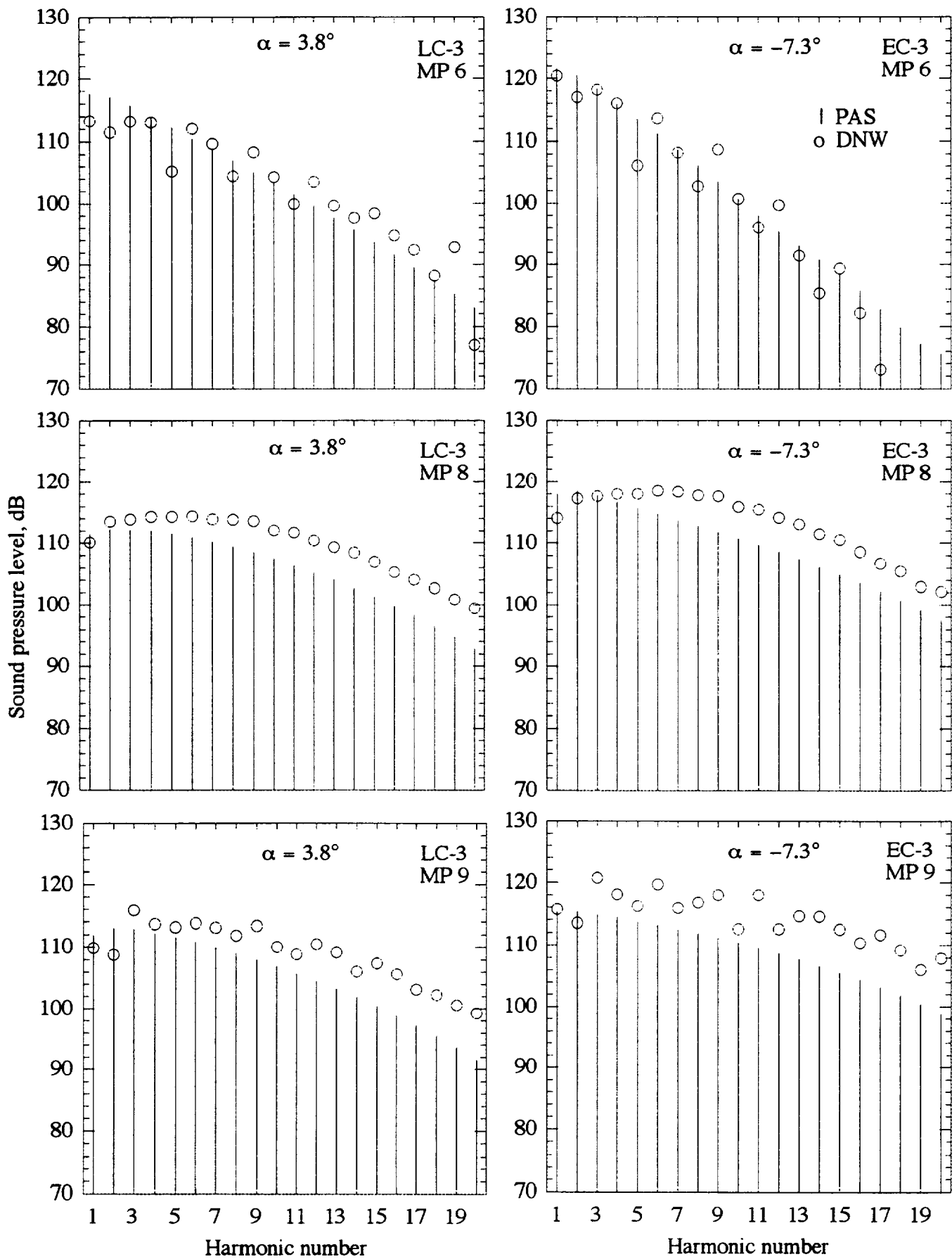


Fig. 30 Comparison of prediction with DNW data at MP6, MP8 and MP9 for the square tip propeller.  $\Omega = 2700$  RPM



**REPORT DOCUMENTATION PAGE**Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE February 1997	3. REPORT TYPE AND DATES COVERED Contractor Report	
4. TITLE AND SUBTITLE A Users Guide for the NASA ANOPP Propeller Analysis System			5. FUNDING NUMBERS NAS1-96014 538-03-13-01	
6. AUTHOR(S) L. Cathy Nguyen and Jeffrey J. Kelly				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lockheed Martin Engineering and Sciences Hampton, Virginia 23666			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-0001			10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA CR-4768	
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Robert A. Golub Final Report				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified/Unlimited Subject Category 71			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>The purpose of this report is to document improvements to the Propeller Analysis System of the Aircraft Noise Prediction Program (PAS-ANOPP) and to serve as a users guide. An overview of the functional modules and modifications made to the Propeller ANOPP system are described. Propeller noise predictions are made by executing a sequence of functional modules through the use of ANOPP control statements. The most commonly used ANOPP control statements are discussed with detailed examples demonstrating the use of each control statement. Originally, the Propeller Analysis System included the angle-of-attack only in the performance module. Recently, modifications have been made to also include angle-of-attack in the noise prediction module. A brief description of PAS prediction capabilities is presented which illustrate the input requirements necessary to run the code by way of ten templates. The purpose of the templates are to provide PAS users with complete examples which can be modified to serve their particular purposes. The examples include the use of different approximations in the computation of the noise and the effects of synchrophasing. Since modifications have been made to the original PAS-ANOPP, comparisons of the modified ANOPP and wind tunnel data are also included. Two appendices are attached at the end of this report which provide useful reference material. One appendix summarizes the PAS functional modules while the second provides a detailed discussion of the TABLE control statement.</p>				
14. SUBJECT TERMS ANOPP-PAS Users Guide; Propeller Analysis System; Propeller noise predictions			15. NUMBER OF PAGES 102	
			16. PRICE CODE A06	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	